AFRL-IF-RS-TR-2004-188
Final Technical Report
June 2004

FEASIBILITY OF SOFTWARE PATCH VERIFICATION

Wetstone Technologies, Incorporated

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-188 has been reviewed and is approved for publication

APPROVED: /s/

WILLIAM E. WOLF
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JUNE 2004	Final Feb 04 – May 04

4. TITLE AND SUBTITLE
FEASIBILITY OF SOFTWARE PATCH VERIFICATION

5. FUNDING NUMBERS
C - FA8750-04-M-0059
PE - 606N727
PR - G11A
TA - 00
WU - 01

6. AUTHOR(S)
Chester Hosmer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Wetstone Technologies, Incorporated
17 Main Street, Suite 237
Cortland New York 13045

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFGB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2004-188

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: William E. Wolf/IFGB/(315) 330-2278/ William.Wolf@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

The goal of this brief effort was to determine the feasibility of developing a process that verifies if critical information system software patches behave as intended and introduce only the specific functionality identified for the patch. Based on our research, examination and experimentation, we have not only determined that it would be feasible to develop such a process, but also that this process and associated technology and standards are desperately needed.

14. SUBJECT TERMS
Software Patch, Verification, Patch Deployment, Patch Authentication and Dissemination Capability, PADC

15. NUMBER OF PAGES
107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1 INTRODUCTION 1

1.1 PURPOSE 1
1.2 BACKGROUND 1

2 PROJECT EXECUTION & FINDINGS 12

2.1 COLLECT PATCHES FROM MAJOR SOFTWARE VENDORS 12
2.2 ANALYZE & EXAMINE COMPLETENESS & CONSISTENCY OF PATCH INFORMATION 26
2.3 CREATE A MATRIX OF "CRITICAL BUT MISSING" PATCH INFORMATION 26
2.4 EXAMINE TOOLS & METHODS FOR CRITICAL SYSTEMS INTERROGATION & PATCH MANAGEMENT 29
2.4.1 Large Scale Interrogation Tools 30
2.4.2 Low Cost Interrogation Tools 31
2.4.3 Case Studies 33
2.4.4 Create Matrix of Attributes that can be Extracted (Automated and through Q&A) About a Critical System 77
2.5 USING SCORING MODELS FOR PATCH RISK ANALYSIS 81
2.5.1 Scoring Model Overview 81
2.5.2 The Math Behind the Models 90

3 CONCLUSIONS 99

4 RECOMMENDATIONS 100

5 REFERENCES 103

List of Tables

TABLE 1 SECURITY FLAW GENESIS 7
TABLE 2 SECURITY FLAW TIME OF INTRODUCTION 8
TABLE 3 SECURITY FLAW LOCATION 10
TABLE 4 SOFTWARE COLLECTION TABLE 14
TABLE 5 SOFTWARE PATCH ATTRIBUTE TABLE 25
TABLE 6 CRITICAL BUT MISSING INFORMATION 27
TABLE 7 CRITICAL SYSTEM ATTRIBUTE TABLE 77
TABLE 8 OVERVIEW OF SCORING MODELS 82
TABLE 9 PATCH MANAGEMENT 88
TABLE 10 VENDOR SERVICES 89
TABLE 11 FAIR ISAAC SCORING MODELS 91

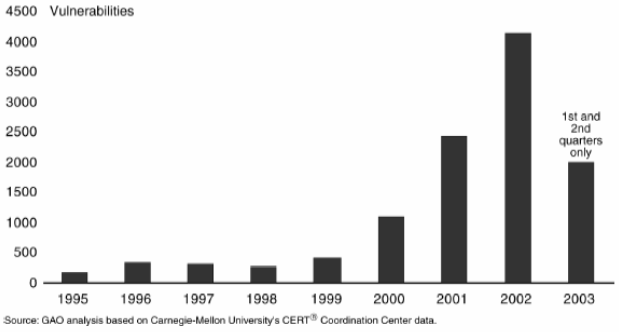
1 Introduction

1.1 Purpose

This Final Technical Report documents all technical work accomplished and reports on the information gained during the performance of contract FA8750-04-M-0059 and its associated Statement of Work (SoW).

1.2 Background

Computer system and software patches have become pervasive in recent years as operating system and application vendors attempt to keep pace in the marketplace with respect to security, functionality, stability, and robustness. Today's complex systems require multifaceted software solutions, while marketplace demands require these solutions in a timely manner. In addition, the increased activity from computer attackers forces vendors to face the difficult challenge of keeping their systems up to date and secure through the distribution of software patches. "Since 1995, over 11,000 security vulnerabilities in software products have been reported. Along with these increasing vulnerabilities, the sophistication of attack technology has steadily advanced. Attacks such as viruses and worms that once took weeks or months to propagate over the Internet now take only hours, or even minutes. In just the past 3 months, two critical and widespread vulnerabilities were identified in products from Microsoft Corporation and Cisco Systems, Inc. Federal agencies were affected by the Blaster and Welchia worms, which exploited the Microsoft vulnerability. The response to these recent events illustrates how federal entities are communicating and coordinating with software vendors and security research groups to combat such attacks. Between 1995 and the first half of 2003, the CERT® Coordination Center5 (CERT/CC) reported 11,155 security vulnerabilities that resulted from software flaws."[i]



Source: GAO analysis based on Carnegie-Mellon University's CERT® Coordination Center data.

"Flaws in software code that could cause a program to malfunction generally result from programming errors that occur during software development. The increasing complexity and size of software programs contribute to the growth in software flaws. For example, Microsoft Windows 2000 reportedly contains about 35 million lines of code, compared with about 15 million lines for Windows 95. As reported by the National Institute of Standards and Technology (NIST), based on various studies of code inspections, most estimates suggest that there are as many as 20 flaws per thousand lines of software code. While most flaws do not create security vulnerabilities, the potential for these errors reflects the difficulty and complexity involved in delivering trustworthy code."[ii]

The process of patch issuance, distribution, application, and validation creates a plethora of problems, and System Administrators and security officers are, in return, challenged with keeping their own systems patched in order to ensure proper operation and maintain the integrity of their operations. Vendors must decide how and when to patch, and how to disseminate information about the availability of the patch, the patch itself, and information about what the patch attempts to accomplish. System Administrators must monitor their communication lines for patch availability, obtain patches, assess whether or not a patch should be applied, apply a patch appropriate to their security and administrative policies, and validate that the patch was effective. Fortunately, organizations and technology have become available to assist in this process.

The Patch Authentication and Dissemination Capability (PADC) program managed by the Federal Computer Incident Response Center is an excellent example of an agency providing patching assistance to organizations. The PADC provides a means for government organizations to have confidence that a patch is authentic and that it works as advertised. Through testing and signed distribution of patches, organizations can have more confidence that a patch is genuine and that it genuinely and appropriately addresses the problem(s) for which the patch was designed. Patch availability and integrity are important elements of an organization's patching methodology and the PADC provides a valuable service that helps to ensure this integrity. The organization is designed to respond very quickly as patches become available and the patch testing performed by PADC supplements the vendor's own testing. This testing is valuable to an organization particularly because PADC can act as an independent third party that is not under the same marketplace demands as the vendors.

However, the PADC process solves only one facet of the problem. Even the most tested patches, whether tested through PADC or by the vendors themselves, may or may not be appropriate for an organization to apply. Therefore, beyond the authenticity of a patch,

organizations must also consider the following questions:

- ✓ Is the patch applicable to our systems?
- ✓ How do we know if this patch will affect our critical systems?
- ✓ How many of our systems will the patch affect?
- ✓ How should the application of multiple patches be made?
- ✓ Is there a specific order in which patches should be applied?
- ✓ How will we know if the patch is effective?

To PADC's credit, the program allows vendors to more quickly identify what patches apply to their own specific configurations. Through a registration process, System Administrators can be notified when a verified PADC patch is available and applicable to their systems. Even with this capability however, a complete patch methodology cannot be developed. Ultimately, a patch's effect on a system is based on specific local configurations, which vary immensely from one organization to the next. An organization like PADC cannot perform tests on adequate differing systems or configurations to cover the basis for the multitude of configuration variations that exist globally. In addition, configurations are often proprietary or sensitive, thus limiting how extensively an external organization can assist in the applicability assessment process.

Organizations must develop a Patching Methodology (PM) that is specific to their own systems and their own security and operating requirements. A PM must consider local policies when assessing patch applicability and a sound method must be developed for patch application. Fundamentally, the problem of patch application is a problem of risk assessment and risk management.

Organizations must consider some of the follow issues when considering their own patching methodologies:

- ✓ How does the patch affect an organization's critical systems?
- ✓ Are there compatibility or interoperability issues between the patch and the local applications or local configurations?
- ✓ Does the patch introduce new vulnerabilities either in the general case or within a specific configuration?
- ✓ What resources will application of the patch consume, will there be any down time, how much will it cost?
- ✓ What happens if the patch fails?
- ✓ What process should be used for patch validation?
- ✓ How will a specific patch or set of patches be deployed?

- ✓ When should a specific patch or set of patches be deployed?
- ✓ What internal testing methodology should be employed
- ✓ What hostile testing should be applied to ensure patch effectiveness?
- ✓ What is the history regarding the effectiveness and security of patches delivered by this organization and how should this be applied to scoring the result?

The answers to these questions are complex, and they are specific to an organization's systems and requirements. Consider the deployment of multiple patches within a system, perhaps from two different vendors for two different applications. In this case, how will an organization determine the order in which the patches should be deployed? An organization may have a difficult time determining if the patches are even compatible with one another. In situations like this, is it possible that improper patch deployment can actually introduce vulnerabilities within a system.

A patch deployment strategy based on risk management must consider these issues and more. For critical systems, levels of risk need to be significantly lower than non-critical systems and may require more complex patch deployment strategies, which in turn are likely to be more costly to the organizations.


CSO
The Resource for Security Executives
cso online.com Home | Magazine | Newsletters | Career | Online Features | Resources | Search
August 2003 CSO Magazine
PATCH AND PRAY
It's the dirtiest little secret in the software industry: **Patching** no longer works. And there's nothing you can do about it. Except maybe patch less. Or possibly patch more.
BY SCOTT BERINATO
Early one Saturday morning in January, from a computer definitely located somewhere within the seven continents, or possibly on the four oceans, someone sent 376 bytes of code inside a single data packet to a SQL Server. That packet—which would come to be known as the Slammer worm—infected the server by sneaking in through UDP port 1434. From there it generated a set of random IP addresses and scanned them. When it found a vulnerable host, Slammer infected it, and from its new host invented more random addresses that hungrily scanned for more vulnerable hosts.

The web site screen shot above illustrates the common place frustration that exists within Information Technology (IT) today regarding patch deployment. It is clear that we have

arrived at a point in time where “patch and pray” is the standard. This may be humorous, however it has become a daily reality for those protecting our critical infrastructures.

WIRED NEWS

Home | Technology | Culture | Business | Politics | Wired Mag | Animation

Text Size: A A A

Windows Users Knocked Off Net By Associated Press

Story location: http://www.wired.com/news/technology/0,1282,59006,00.html

02:03 PM May. 27, 2003 PT

WASHINGTON -- Microsoft withdrew a security improvement for its flagship Windows XP software after it crippled Internet connections for some of the 600,000 users who installed it. Microsoft officials said Tuesday the update -- which had been available as an option since Friday on its "Windows Update" website -- apparently was incompatible with popular security software from other companies, such as Symantec.

Microsoft said Internet connections failed immediately for an unspecified number of more than 600,000 computers using Windows XP who downloaded and installed the update. Consumers could reconnect only by removing the update, which promised to improve reliability for types of secure Internet connections commonly used by corporations.

[iii]

Making matters worse are the all too common-place errors that occur after patches have been employed that cripple our information systems. These difficulties have caused System Administrators to develop elaborate schemes to back out failed patches or switch over to backup infrastructures (if they can afford to have such luxuries) on the fly in order to support those who rely on their critical systems.

Exactly how big is the problem then? “A May 2002 report prepared for the National Institute of Standards and Technologies (NIST)(1) estimates the annual cost of software defects in the United States as $59.5 billion.”[iv]

“The root cause of these astronomical costs is the increasing complexity of today’s software. In the early days of computing, there were strict memory limits that inherently kept complexity in check by limiting code size. As these memory requirements have disappeared and processor performance has improved, the requirements for software have fundamentally changed. Modern server-side applications such as operating systems, application servers, and databases usually contain hundreds of thousands, if not millions, of lines of source code.

An unfortunate repercussion of today’s world of networked computing contributes a second factor to the economic impact of software quality. Once software is operating in a networked environment, virtually every bug in that software becomes a potential security hole. If there is any way for an outsider to trigger the particular code path to the bug, that bug is now at least a DoS (Denial of Service) attack waiting to be discovered, if not worse. The occurrences of such security attacks have grown exponentially over the past

several years, and more than eight out of ten corporations in the United States were victims of security breaches during the past year."[v]

How did we arrive at this place where software patching has become so common place and such an arduous process? In order to answer this question, we must first examine what a security vulnerability or flaw is. In the simplest terms a security vulnerability is an intentional or unintentional bug that is introduced in an area of a software system that can cause the system to violate a basic security principal. Our basic going in position regarding security vulnerabilities is that they are unintentional, however, the risk does exist that some percentage of these faults have been intentionally placed in a software code base for the specific purpose of attacking them at a later time.

In order to illustrate the taxonomy of flaws that exist we have modeled and modernized the following tables. The tables are modeled after a 1994 ACM Computing Survey that attempted to define a security flaw taxonomy.

Table 1 Security Flaw Genesis [vi]

	Category	Type	Risk
Security Flaw Genesis	Intentional	Trojan Horse	Unauthorized Root Access, Trigger destruction of information, background information processing, denial of service or attack
		Trap door	
		Time bomb	
		Covert channel	Intermittent or continuous flow of unauthorized information
		Spyware	Malicious surveillance of user activity, including password capture
	Unintentional	Buffer overflow	Unintended root access during vulnerability exploit or attack
		Inadequate authentication measures	Potential unauthorized user access through exploitation
		Validation error	Exploitation of yet unknown vulnerability
		Covert channel	Potential Intermittent or continuous flow of unauthorized information
		Boundary condition error	

"Both malicious flaws and non-malicious flaws can be difficult to detect, the former because they have been intentionally hidden and the latter because residual flaws may be more likely to occur in rarely invoked parts of the software. One may expect malicious code to attempt to cause significant damage to a system, but an inadvertent flaw that is exploited by a malicious intruder can be equally dangerous"[vii]

Based upon the security flaws listed above and continuing to build upon the work of the aforementioned ACM Survey, we need to examine the time of introduction of such security flaws. Of particular importance related to this study is the examination of

security flaws and their associated risks when the flaws are introduced as a result of the patch process.

Table 2 Security Flaw Time of Introduction[viii]

	Category	Type	Risk
Time of Security Flaw Introduction	During Development	Poor Requirements Specification	Fundamental security flaws exist that can be exploited anytime during the lifecycle of the software
		Poor Design	
		Implementation Errors	Common programming errors can be exploited during the lifecycle. These flaws can be "tested out" if the software system is "open source" then peer review may be effective. In proprietary software implementations "black box" testing and reverse engineering are required to uncover flaws, unless the source code is stolen or inadvertently released.
		Poor Testing Methods	Security flaws that might have been typically identified get released to customers
	During Feature Update	Implementation Errors	Same as above, but typically isolated to areas modified and common modules
		Inadequate Testing	Testing shortcuts that focus only on added features and ignore security or regression testing can be catastrophic
	During Patch Updates	Implementation Errors	Same as above, but more likely due to the extreme market pressure applied to fix the problem. Typically patches are delivered to the marketplace within 72 hours of notification of the security flaw. This yields both the danger of poorly conceived patches as well as the risk of the introduction of additional security or operational flaws that may be more serious than the original problem.
		Symptom based patches	Attention on the symptom, may miss other similar or identical flaws in other areas of the code base.
		Inadequate Testing	Same as above with the added time-pressure that calls for even more testing short-cuts

The final table in the series identifies the specific location of the security flaw. This helps to identify what type of impact or risk does the vulnerability pose to our critical information systems.

Table 3 Security Flaw Location[ix]

	Location	Type	Risk
Software	Operating System	BIOS	Security flaws in any of these areas can have catastrophic impacts on critical information systems. The OS and the related services provide the underpinnings for critical system operation.
		System Boot	
		Kernel	
		Device Drivers	
		Memory Management	
		Communication Stack	
		Process Management	
		Identification / Authentication	
		Access Control User Management	
		Directory	
		Cryptographic	
		File Management	
	Application	Privileged	Same as Operating System Above
		Unprivileged	Same as Operating System Above if they can cause users to make critical mistakes that lead to preventable errors. i.e. inadvertently launching malicious code that was an attachment to an e-mail.
		Security	Same as Operating System above with the added risk of a false sense of security. Security applications such as virus protection services, malware and spyware detection, 3rd party cryptographic and certificate services, host based intrusion detection and firewall technologies are today common place. Security flaws in any of these applications can have serious ramification. First, these programs typically are privileged and secondly are "blindly" relied upon by users in most cases.
	Support	Privileged Utilities	Same as Security Applications above
		Unprivileged Utilities	Same as Unprivileged Applications Above

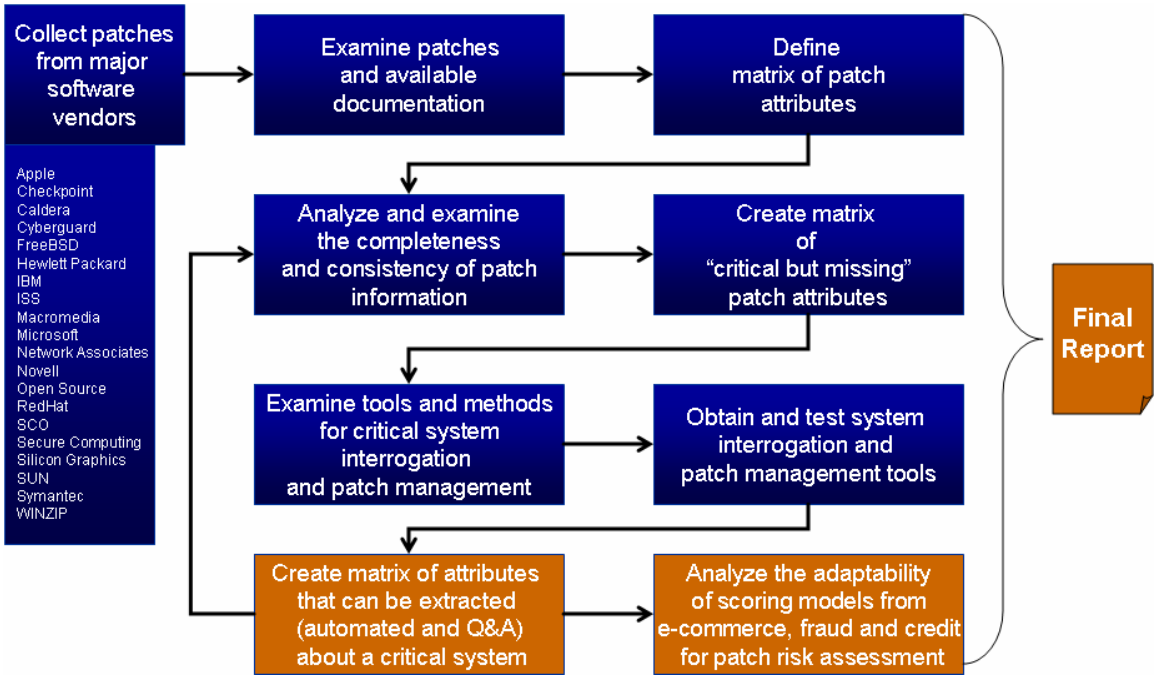
By examining the categories of security flaws, the time of their introduction into a system and the specific software components affected provide valuable information regarding the risks associated with the flaws along with the threats that they may pose to our critical information systems. Furthermore, when applying patches that attempt to correct flaws related to the triad (category, time of introduction and location) can provide significant insight into the patch deployment strategy and risk modeling.

In addition, based on the complexity of modern critical system infrastructures we must urgently devise new methods for the assessment of risks associated with patch deployments and demand more from those providing the patches as well as the underlying software systems that they provide.

According to Government Accounting Office, (GAO) Testimony “Another critical step is to test each individual patch against various systems configurations in a test environment before installing it enterprise-wide to determine any impact on the network. Such testing will help determine whether the patch functions as intended and its potential for adversely affecting the entity’s systems. In addition, while patches are being tested, organizations should also be aware of workarounds, which can provide temporary relief until a patch is applied. Testing has been identified as a challenge by government and private-sector officials, since the urgency in remedying a security vulnerability can limit or delay comprehensive testing. Time pressures can also result in software vendors’ issuing poorly written patches that can degrade system performance and require yet another patch to remediate the problem. For instance, Microsoft has admittedly issued security patches that have been recalled because they have caused systems to crash or are too large for a computer’s capacity. Further, a complex, heterogeneous system environment can lengthen this already time-consuming and time-sensitive process because it takes longer to test the patch in various systems configurations.”[x]

2 Project Execution & Findings

Based on the problems and issues described above, we developed the following diagram to illustrate the process we defined to examine the issues of patch deployment for critical information systems.



2.1 Collect Patches from Major Software Vendors

During this phase of the project we first decided to select a set of vendors that would be a representative sample of those issuing software patches that would have relevance to critical information systems. In addition, we attempted to select patches that spanned operating systems, applications and security software. In addition to collecting the patches themselves we also collected all pertinent patch documentation, examine user group boards for additional information. It should be pointed out that obtaining these patches in some cases was quite simple in other cases very difficult. Many vendors (of operating systems and applications that we did not have a license for) were very difficult to convince that we should be allowed to examine them.

Vendor List		
Apple	**Checkpoint**	Cisco
Caldera	**Cyberguard**	Free BSD
Hewlett Packard	**IBM**	ISS
Macromedia	**Microsoft**	Network Associates
Novell	**Open Source**	Red Hat Linux
Santa Cruz Operations	**Secure Computing**	Silicon Graphics
Sun	Symantec	Winzip

Cisco Systems, for example, refused to provide us with any specific patches. The list above shows the vendors whose patches we attempted to collect during the effort.

The next step in our approach was to examine the collected patches (along with any accompanying documentation, alert notifications and support information) and create a matrix of attributes extractable from the patch distributions. This process included examining the patch release itself, to determine attributes such as size, number of affected files and difference between currently existing software components. This process proved to be too ardous without specialized software that could automatically compare the current modules with the proposed patched module. Further analysis revealed that in order to do this effectively, either a specialized software application would need to be devised (with personality modules for each vendor), or additional information or tools may be required from the vendors releasing the patches. The result of this task was the following software patch collection table and the subsequent patch attribution table.

.

Table 4 Software Patch Collection Table

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
WinZip Computing	WinZip 9.0	Windows	3/3/2004	9	Buffer Overflow	Critical	Double clicking on certain files that will invoke winzip. If this is done with an invalid file, it could cause a buffer overflow.	2318K	Cannot Determine	Installer
Apple	GdbFixes_OSXS.tar	WebObjects 4.0.1 on Windows NT and Solaris	10/18/2000	4.5	Bug FIx	N/A	Fixes problem with print object command	30KB (zipped)	Cannot Determine	Varies with operating system
Apple	SecUpd2004-02-23Pan.dmg	Mac OS X 10.3.1 or later - Client only Security Update for Panther Client 1.0	2/23/2004		Cannot determine	N/A	Addresses a number of security enhancements	1261KB	Cannot Determine	Manual
Apple	SecurityUpd2003-06-09.dmg.bin	Security update for 10.2.8 Server Mac OS X 10.2.6 or later	3/6/2009		Cannot determine	N/A	Addresses a number of security enhancements	1035KB	Cannot Determine	Manual
Apple	QT412Patch.exe	Quicktime Patch		4.1.2	Buffer overflow	N/A	Buffer overflow correction	33KB	Cannot Determine	Manual
Apple	SecUpdSrvr2004-01-26Jag.dmg	Security update 10.2.8 Server	1/26/2004		Cannot determine	N/A	Addresses a number of security enhancements	7135KB	Cannot Determine	Manual
HP	PHSS_29813	HP-UX Network Node Manager		3.2	Memory Leak and other fixes	Critical	Variety of fixes	14520 KB	Documented	Manual
HP	OMNIBACK_00109.EXE	Omniback II on Windows		4.1	Bug FIx	N/A	Variety of fixes	3400KB	Documented	Installer
HP	OPSPSOL_00002	Openspool on Sun Solaris		b.01.60	Bug FIx	Non critical	Addresses 3 non critical year 2000 issues	12850 KB	Documented	Manual
HP	PHSS_29683	Performance Agent on HP-UX		c.03.71.00	Bug FIx	N/A	Variety of fixes	180KB	Documented	Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
HP	SDSK_00034.EXE	Openview Service Desk on Windows		4.5	This is a Service Pack	N/A	Large variety of fixes	25457 KB	Documented	Manual
IBM	C44CCNA.exe	Lotus Workflow	5/23/2002	Public Fix Pack 1	Bug FIx	N/A	Fixes a variety of problems	443634 bytes	Cannot Determine	Installer
IBM	4.5.1-TIM-0005	Identity Manager		4.5	Bug FIx	N/A	Fixes a variety of problems	48868 (zipped)	Documented	Manual
IBM	4.5.6-TIM-0005	Tivoli Web Services Manager	3/5/2004	4.5.6	Bug FIx	N/A	Fixes a variety of problems	48868 (zipped)	Documented	Manual
IBM	Patch QQQ	LearningSpace 5.01 on Windows	10/24/2003	5.01	Bug FIx	N/A	Contains a fix	97K zipped	Documented	Installer
IBM	WSAM 2.1.2	WebSphere Studio Application Monitor for z/OS and OS/390	2/8/2004	Patch version 07	Bug FIx	N/A	Contains fixes	8937K zipped	Documented	Manual
Microsoft	Windows2000-KB823182-x86-ENU.exe (MS03-041)	MS Windows	10/15/2003	1.2	Remote Code Execution	Critical	The Authenticode capability in Microsoft Windows NT through Server 2003 does not prompt the user to download and install ActiveX controls when the system is low on memory, which could allow remote attackers execute arbitrary code without user approval.	360K	Cannot Determine	Installer

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Microsoft	Windows2000-KB824146-x86-ENU.exe	MS Windows	10/10/2003		Three new vulnerabilities, the most serious of which could enable an attacker to run arbitrary code on a user's system. Buffer Overrun-Denial of Service-Code Execution	Critical	Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly access services on another computer. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.	917K	Documented	Installer

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Microsoft	Windows2000-KB826232-x86-ENU.exe	MS Windows	10/15/2003	2	Remote Code Execution	Critical	A security vulnerability exists in the Microsoft Local Troubleshooter ActiveX control. The vulnerability exists because the ActiveX control (Tshoot.ocx) contains a buffer overflow that could allow an attacker to run code of their choice on a user's system. Because this control is marked "safe for scripting", an attacker could exploit this vulnerability by convincing a user to view a specially crafted HTML page that references this ActiveX control. The Microsoft Local Troubleshooter ActiveX control is installed as a default part of the operating system on Windows 2000.	330K	Cannot Determine	Installer
Microsoft	Windows2000-KB828749-x86-ENU.exe	MS Windows	11/11/2003	1.2	Remote Code Execution	Critical	A security vulnerability exists in the Workstation service that could allow remote code execution on an affected system. This vulnerability results because of an unchecked buffer in the Workstation service.	329K	Cannot Determine	Installer

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Microsoft	Windows2000-KB824141-x86-ENU.exe	MS Windows	10/15/2003	4.1	Local Elevation of Privilege	Important	An attacker who had the ability to log on to a system interactively could run a program that could send a specially-crafted Windows message to any applications that have implemented the ListBox control or the ComboBox control, causing the application to take any action an attacker specified. This could give an attacker complete control over the system by using Utility Manager in Windows 2000.	3474K	Cannot Determine	Installer
Novell	fgwapi5.exe	GroupWise API Gateway 4.1	6/12/2003	2	Bug FIx	N/A	Fixes 5 application problems (see documentation)	735604 bytes	Documented	Manual
Novell	in42sp2.exe	InForms 4.2 Service Patch 2	3/26/1998	1	Bug FIx	N/A	Fixes 9 application problems (see documentation)	7338542bytes	Documented	Installer
Novell	netmail310log.exe	NetMail 3.10 on Windows	11/13/2003	1	Logging violations	N/A	Provides file and instructions to help administrators setup logging for NetMail 3.10.	7479 bytes	Documented	Installer and Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Novell	notes51.exe	GroupWise 5 NT Gateway for Lotus Notes patch 1	6/8/2000	1	Bug FIx	N/A	Fixes 7 application problems (see documentation)	45561 71 bytes	Documented	Installer
Novell	imspmfix.nlm	NetWare	3/8/2002	1	Bug FIx	N/A	Postmaster Fix utility	93664 bytes	Documented	Manual
Secure Computing	GNT50pt1.zip	Gauntlet 3.0 and 5.0 for NT	4/1/1999	V3.0, v5.0, patch 1	Bug FIx	N/A	Fix memory corruption and leak and two other fixes (see documentation)	79K (zipped)	Cannot Determine	Manual.
Secure Computing	PA_30003.exe	SafeWord PremierAccess Version 3.0 on Windows or Unix	4/15/2002	3.0.0.03	Remote exploitation, buffer overflow	N/A	*Addresses CERT Advisory CA-2002-06 Vulnerabilities in Various Implementations of the RADIUS Protocol	191K (zipped)	Documented	Manual
Secure Computing	PA31101.exe	SafeWord PremierAccess Version 3.1 on Windows or Solaris	12/6/2002	3.1.1.01	Access control list bug	N/A	Roles returned from the Authentication Broker were not being used to process ACL entries. Phoenix device enrollments could not proceed because the Phoenix CSP could not be detected.	214K (zipped)	Documented	Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Secure Computing	Sidewinder50 Patch4	Windows	11/3/1999	5.0.0.04	Bug Fix	N/A	Corrects a number and variety of problems/issues Stated as “Cloning and System Resource Enhancements Patch)	1741K (zipped)	Cannot Determine	Installer
Secure Computing	Sidewinder V5.0.0.04for Unix.txt	Unix	Assume same as above	5.0.0.04	Bug Fix	N/A	Corrects a number and variety of problems/issues Stated as “Cloning and System Resource Enhancements Patch)	?	Cannot Determine	Manual
CheckPoint	DNS_SSL_VLAN_HF_Nokia.tgz	Unix	8/6/2002		Buffer Overflow	N/A	Fix for a buffer overflow in the bind DNS resolver library	13,624K (zipped)	Cannot Determine	Installer
CheckPoint	security_server_hotfix_cpsc.zip	Unix	2/4/2004		Bug Fix	N/A	A vulnerability causes the server to crash under certain cirumstances	12K (zipped)	Cannot Determine	Manual
CheckPoint	OpsecSdkNgFp2HotFix.ipso.tar.gz	Windows/Solaris/Linux	Aug-02		Multiple Vulnerabilities	N/A	Corrects a number of vulnerabilities	2,012 KB (zipped)	Cannot Determine	Manual
CheckPoint	NG_FP2_SS_HF_Nokia.tgz	Unix	8/29/2002		Bug Fix	N/A	Contains fixes and improvements	29J (zipped)	Cannot Determine	Manual
CheckPoint	NG_FP2_Alerts_HF_nokia.tgz	Unix/Windows/Solaris/Linux	Apr-02		Bug Fix	N/A	Contains two fixes	1,228K (zipped)	Cannot Determine	Manual
CyberGuard	global_orders.tar.encr	Unixware	5/16/2003	5.1	Bug Fix	N/A	Contains improvements	26,412 KB	Cannot determine	Cannot determine

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
ISS	Heap Overflow	Multiple products	9/5/2001		Heap Overflow	N/A	ISS RealSecure and BlackICE product lines comprise of network-based and host-based intrusion detection and prevention systems. Many of these products contain a heap-based buffer overflow vulnerability that can be triggered by a specially crafted Server Message Block (SMB) request.		Cannot Determine	Installer
ISS	RSNetSnsr70_MU_22_12.ini	Proventia XPU	3/9/2004	7.0 sensor	Cannot determine	N/A		15468 bytes	Cannot Determine	Cannot determine
ISS	BISentryMSSetup.exe	RealSecure Sentry	12/14/2001		Cannot determine	N/A		6854KB	Cannot Determine	Installer
ISS	rscli7.0.2003[1]	RealSecureWorkgroupManager Solaris 7.0	1/19/2004		Cannot determine	N/A		13399552 bytes	Cannot Determine	Installer
ISS	RSNetSnsr70_MU_22_12.ini	X-Press	3/9/2004		Cannot determine	N/A		15468 bytes	Cannot Determine	Installer
Macromedia	dwmx2004_701update_en.exe	Dreamweaver MX2004 Windows/Macintosh		7.0.1	Bug Fix	N/A	Fixes known problems	20,631 KB	Cannot Determine	Installer
Network Associates	4320eng.exe	Engine-only Superdat File (Intel)		4.3.20	Cannot determine	N/A	Engine Update	3,515 KB	Documented	Installer
Network Associates	EPO2xP2.Exe	McAfee ePolicy Orchestrator 2.X Patch 2	7/31/2003	Version 2.0, 2.5, and 2.5.1 Patch 2	Obtain password/arbitrary code execution	N/A	Includes 2 hotfixes and 1 patch	2,524 KB (zipped)	Cannot Determine	Manual
Network Associates	EPO3002.zip	McAfee ePolicy Orchestrator 3.0 Patch 2	7/30/2003	Version 3.0 Patch 2	Obtain password/read arbitrary files	N/A	Fixes two security vulnerabilities	1,716 KB (zipped)	Cannot Determine	Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Network Associates	EPO25113.zip	Patch 13 for ePolicy Orchestrator 2.5.1	1/30/2004	Version 2.5.1 Patch 13	HTTP POST Buffer Mismanagement Vulnerability/Execution of arbitrary code	N/A	Fixes three security vulnerabilities	2,481 KB (zipped)	Cannot Determine	Manual
Network Associates	EPO3013.zip	Patch 3 for ePolicy Orchestrator Agent 3.0 Service Pack 1	1/28/2004	Version 3.0.1 Patch 3	HTTP POST Buffer Mismanagement Vulnerability	N/A	Fixes stated vulnerability	1,756 KB zipped	Cannot Determine	Manual
Open Source	507up2_vol.tar	SCO OpenSever(TM) Release 5.0.7 Update Pack 2	1/30/2004	Release 5.0.7 Update Pack 2	Bug FIx	N/A	Upgrades product	34,785 (zipped)	Cannot Determine	Live Update
Open Source	lib/checkpw.c.orig	FreeBSD	12/28/2003	2.1.17	Cannot determine	N/A	Upgrade	5.2KB	Cannot Determine	Manual
Open Source	linux-2.4.22-5rxv-source-2087.patch.bz2	RedHat		Red Hat Linux 9 kernel interactivity patch	Cannot determine	N/A	Upgrade	21KB	Cannot Determine	Manual
Santa Cruz	cyrus-imapd-2.1.10-10ol.i586.rpm	SCOoffice Mail Server 2.0 Cyrus Update for OpenLinux		SCOffice Mail Server 2.0 onCaldera Open Linux 3.1.x Volution Messaging Server on aldera Open Linux 3.1.x	Buffer Overflow	N/A	Upgrade and security fixes	2,711 KB		Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Santa Cruz	507up2_vol.tar	SCO OpenServer Release 5.0.7 Update Pack 2		Release 5.0.7 Update Pack 2	Bug Fix	N/A	Addition of new features	34,785 KB	Cannot Determine	Live Update
Santa Cruz	SCOofficeMail Connector_2.0.55.zip	SCOoffice Mail Connector & Address Book for Microsoft(r) Outlook(r)	11/26/2003	build 55 (2.0.55)	Bug Fix	N/A	Bug fixes	3,566 KB (zipped)	Cannot Determine	Installer
Silicon Graphics	patch4713.tar	BDSpro 2.4 software	8/8/2002	4713	Daemon security vulnerability	N/A	Contains bug fixes	110KB (zipped)	Cannot Determine	Manual
Silicon Graphics	patch4799.tar	SG0004799 to IRIX 6.5 SG0004799 to IRIX 6.5	9/27/2002	4799	Bug Fix	N/A	Contains bug fixes	3660KB (zipped)	Cannot Determine	Manual
Silicon Graphics	patch4915.tar	patch SG0004915 to IRIX 6.5	3/27/2003	4915	Bug Fix	N/A	Contains bug fixes	7,990 KB (zipped)	Cannot Determine	Manual
Silicon Graphics	patch5065.tar	patch SG0005065 to Samba 2.2.8 for IRIX	4/8/2003	5065	Bug Fix	N/A	Contains bug fixes	1,360 KB (zipped)	Cannot Determine	Manual
Silicon Graphics	patch5313.tar	patch SG0005313 to DCE/DFS1.2.2c on IRIX 6.5.9m or 6.5.9f and above	9/25/2003	5313	Bug Fix	N/A	Contains bug fixes	10,050 KB (zipped)	Cannot Determine	Manual
Sun	J2SE_Solaris_2[1].5.1_Recommended.tar.Z	Solaris 2.5.1	2/19/2003	2.5.1	Bug Fix	N/A	Cluster Patch (multiple patches) Contains updates	14,827 (zipped)	Cannot Determine	Manual
Sun	T116309-01.zip	Security T-Patch	11/11/2003	Patch-ID# 116309-01	Bug Fix	N/A	Contains a bug fix, and is a temporary intermediate fix	249KB (zipped)	Cannot Determine	Manual
Sun	117073-01.tar	StarOffice/StarSuite 7_x86 (Solaris)	3/5/2004	Patch-ID# 117073-01	Bug Fix	N/A	Contains several bug fixes	36,418 KB (zipped)	Cannot Determine	Manual

Vendor Name	Patch Name	OS or Application	Date of Issue	Version Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Affected Files	Deployment Method
Sun	ControlStation-All-OS-Update-4.0.pkg	1.x versions of the Sun Cobalt Control Station	10/9/2003	Update 4	Remote exploit-Attacker could gain root access	N/A	Contains security fix	538,416KB	Cannot Determine	Cannot determine
Sun	zebra-0.91a-8.7.2.i386.rpm	Zebra 0.91a-8.7.2	12/15/2003	8.7.2	Remote exploit/Denial of service	N/A	Contains 2 security fixes	965KB	Cannot Determine	Cannot determine
Symantec	esm(for Lunux).tpk	Symantec Enterprise Security Manager for Windows and Unix Modules		Security Update 18		N/A	Contains several security fixes	18,710 KB	Cannot Determine	Live Update
Symantec	apache_vulnerable_cgi_scripts.pol	Intruder Alert 3.6 Apache	7/30/2002		c - Vulnerable CGI Scripts Policy	N/A	A security administrator uses this policy to track usage of cgi scripts, thereby watching for the misuse of cgi scripts via remote access	31KB	Cannot Determine	Cannot determine
Symantec	Available only through Live Update	Symantec NetRecon 3.6 Security Update 15	3/25/2004	Symantec NetRecon 3.6 Security Update 15	Information disclosure/denial of service	N/A	Contains several security fixes	Cannot determine	Cannot Determine	Installer

Table 5 Software Patch Attribute Table

Information	Category	Description
Vendor Name	General	The name of the vendor releasing the patch
Patch Name	General	The name of the patch assigned by the vendor
Product Name	General	The name of the specific product being patched
Version Number	General	Version number of the specific patch associated with the product name
Date of Issue	General	Date when the vendor first issued the patch
Date of Update	General	Date of any updates to the patch
Patch Description	Security	Description by the vendor of the nature of the patch
Impact Potential	Security	Security or operational impact of the patch. In other words what problem (security or operational) does the patch attempt to address or fix.
Rating	Security	The severity rating of the underlying problem that this patches addresses. This could also include recommendations as to the urgency or time related nature of the suggested patch.
Size of Patch	Risk	The size of the patch in bytes typically is reported as the size total size of the installable package.
Affected files	Risk	A small number of vendors supply detailed list of what specific files are being added, replaced or deleted as part of the patch
Patch Deployment Method	Risk	In most cases this is an executable file that you run in order to apply the appropriate patch. However, in some cases a single or set of files will be provided and the system administrator will manually replace these files or some rare cases make other system file edits.
Patch Security	Security	Some patches containing embedded security information such as digital signatures and digital timestamps that allow vendors to verify the integrity and authenticity of a patch.

It seems obvious that that the information typically provided by software vendors represents a small amount of the information needed by critical System Administrators and managers regarding the content of a patch. Most vendors have remained steadfast in the limited information they provide related to each hotfix, QFE, security update, or service pack that describes the set of vulnerabilities that these fix. One typical reason cited is information related to vulnerabilities distributed with patches will ultimately end up in the hands of potential adversaries that could, and do, exploit this information. This provides a significant dilemma for software vendors at a very crucial time when software patches or security updates are delivered. In almost all cases the software vendors are assuming and sometimes even requiring "*blind adherence*" to the software update process by their customers. This assumption may be difficult for those operating critical infrastructure systems to swallow due to the unknown and non-quantifiable risks that may be posed by this "*blind adherence"* strategy and assumption.

2.2 Analyze & Examine Completeness & Consistency of Patch Information

During this task we compared information across patches provided by different vendors and patches of differing types (i.e. operating system vs. application patches) in order to assess the consistency and completeness. After an initial assessment, we quickly determined that the diversity of patch releases across vendors and categories makes it impossible to examine this collectively. The two key elements that led us to this conclusion are:

1. The information that is provided regarding patches is so limited (see Table 5 Software Patch Attribute Table) that additional research is necessary for almost every case in order to assess risk and relevancy for critical system deployments.

2. Understanding the genesis, time table and location of the bug or security hole is rarely if ever disclosed (see tables: Table 1 Security Flaw Genesis, Table 2 Flaw Time of Introduction, and Table 3 Security Flaw Location).

2.3 Create a Matrix of "Critical but Missing" Patch Information

During this task we identified the "critical but missing" software patch information. We must evaluate the efficacy of patch deployments to our critical infrastructure and assess the following critical questions:

1. Should we apply this software patch
    a. Tradeoff analysis of the security risk vs. the risk of applying a flawed patch.

2. When should we apply the patch
3. What specific systems should we apply the patch too
4. What is the test plan and procedures that should be associated with this patch
5. What type of fail-safe or fail-over should be applied
6. What will the cost of applying the patch be

Based on these critical questions we have developed the following matrix of critical but missing patch information that we feel is necessary in order to effectively answer the questions above.

Table 6 Critical but Missing Information

Critical But Missing Patch Information	Category	Description
Detailed Patch Information	Vulnerability	Today most vendors only offer surface level information regarding patches. They typically do not provide information related to the root cause, genesis, time of introduction or even specific location of the fault. We believe this information is essential in order to determine a risk of the patch. This information must include as a minimum the following: 1. Genesis of the software flaw a. Requirements flaw b. Design flaw c. Implementation flaw d. Previous software maintenance flaw 2. Flaw root cause a. Buffer Overflow b. Memory leak c. Boundary condition d. Logic error e. Race condition f. Authentication error g. Validation error h. Inadequate testing i. Intentionally malicious code j. etc. 3. Source code (snippet) of flaw and proposed fix 4. Location of fault a. Kernel b. Process management

Critical But Missing Patch Information	Category	Description
		c. Memory management d. File management e. Communication stack f. Security stack 5. Time of introduction a. Product version number from – to b. Date of introduction
Size of proposed patch	Patch Details	Understanding the size and scope of the proposed fix can have a direct bearing on both risk assessment as well as testing requirements. The specific details that are needed included: 1. During patch deployment the number of files modified, added and removed. 2. Number of lines of code modified 3. Number of other changes (i.e. registry entries or other data files)
Timetable of patch deployment	Corrective Action Details	In order to assess risk some information is required to understand the process that the vendor has conducted in order to a fix a vulnerability. To this end both calendar time and resources applied to each step would be helpful metrics. These metrics need to be included for: 1. Discovery of the vulnerability (start date) 2. Analysis duration and resources 3. Implementation duration and resources 4. Testing duration and resources
Testing procedures	Validation Details	In order to both estimate and mitigate risks associated with a new patch deployment, some information is required to understand the testing process employed by the vendor. Since in most cases the vendor and the vendor only knows intimate details about the software and testing processes, this information is only available from the vendor. 1. Total number of configurations tested 2. Validation against what attack methods (this is available in some cases) 3. Extent of operational testing of patches 4. Any problems discovered during patch installation under these test conditions 5. How long does it take to apply the patch? 6. Does the system have to be rebooted?

Critical But Missing Patch Information	Category	Description
		7. What end user testing is recommended 8. What precautions are recommended 9. Specific risks associated with applying the patches 10. Other vendor recommendations
Personnel that applied the patches	Patch validation	In order to assess the risks of patch deployment information pertaining to the personnel that crafted the patch is required. These risks include competency, oversight, security and care that was applied. The following general categories of information is suggested. 1. At what locations did the patch generation and testing occur (i.e. Redmond, WA) 2. Skill level of those involved with the patch 3. Oversight skill level 4. Quality methods employed (i.e. testing, code walk-through, independent lab validation) 5. Who signed off and what was the criteria of the sign off 6. Authentication and validation steps taken, i.e. one-way hash, digital signatures, digital timestamps etc.

2.4 Examine Tools & Methods for Critical Systems Interrogation & Patch Management

During this stage of the effort we examined current methods and practices for managing patches and interrogating critical systems.

When examining critical systems for extractable information, we first examined currently available software products that will interrogate your network and report back on the configuration of the target system(s). There are a wide range of options from which to choose, and an even wider expanse of what you may pay for the application, depending on your operating system. Some applications purport to track as few as 50 and up to tens of thousands of users. Additionally, the tasks each application can perform vary greatly. Generally, we found the largest selection of tools among Windows operating systems. However, users of IBM mainframes, Linux, Novell and Sun certainly have excellent options, albeit fewer.

We examined several of the specific tools in order to report on the information that we can glean from using them, specifically, what information will be useful to help us determine what attributes a target critical system has that would be valuable knowledge in deciding whether or not to apply a particular vendor patch. We have categorized the tools into two broad categories Large Scale and Small Scale Interrogation Tools.

2.4.1 Large Scale Interrogation Tools

The more robust tools, or high-end interrogation tools, perform many functions in addition to providing the attributes we are looking for in determining whether or not to apply a patch to a system. Additionally, they perform the functions across a network and store the information centrally. These applications may or may not distribute patches. Below we briefly describe some of the tools and technologies that we examined.

Novell's **ZENworks** for example, will not only allow System Administrators to manage servers over the network, but it will manage and query various server operating systems on the network, including Linux, Solaris, Windows and NetWare. The ZENworks Patch Management software will deploy patches to the "appropriate" target systems. In speaking with a Novell system engineer, the way this system works is that the software performs a "software inventory" of all systems (mainly desktop machines as opposed to servers). The application gathers patch information from PatchLink, a service that gathers patches from all vendors. The end user of ZENworks Patch Management becomes a subscriber of the PatchLink service and all new patches are automatically downloaded to the Patch Management station, and in turn are automatically disseminated to nodes on the network based on the inventory software. It makes no fine-grained assessment as to whether or not the patch is safe, i.e., what registry entries may be affected, whether or not this system is a vital component in the network, etc.

Patch Link Corporation provides its own patch management software, **PatchLink Update**. This fully Internet-based software will also deploy patches over multiple operating systems, Microsoft, UNIX/Linux, Novell NetWare, and MacOS X. The inventory feature of this software, in addition to examining each node for installed software, also takes an inventory of hardware and drivers in the target infrastructure, of which patches are missing. This compilation of information is then archived. Based on this query, which they call "*Patch Fingerprinting Technology*", (which is patent-pending), patches are disseminated automatically across the network to the systems it deems appropriate. There is also a manual component to this software allowing an administrator to create computer groups and patch deployment policies based on criteria determined by that administrator.

IBM's **Tivoli Configuration Manger** is an application which will deploy patches across multiple, geographically dispersed operating systems, such as IBM AIX 4.3.3, 5.1, Solaris 7 and 8, Windows 95/98/NT/2000 Pro/2000 Server, Red Hat, SuSE, Novell Netware, OS/2 Warp Server, Palm OS and PalmPC, to name only a few. This application inventories hardware and software information across the enterprise, and stores this information in a database. Based on an assessment, the administrator determines which patches the software is to deploy en masse to which nodes, and the Configuration Manager will apply those patches over the enterprise.

Absolute Track by Absolute Software Corporation, performs remote tracking of PC configuration and installed software. It will query for operating system, service packs for operating systems, and version information of installed software. This program compiles information dynamically, rather than on demand. It is networked to interrogate remote systems.

PC-Duo Enterprise Inventory Management by Vector Networks Ltd. appears to track only a predefined dataset of software and hardware items over the network. It is configurable to manually add additional components and software packages.

2.4.2 Low Cost Interrogation Tools

On the lower end of the spectrum are tools that provide local system information; i.e., they are installed on the workstation for which information is sought. The tools in this category range from the robust (providing detailed information about the target system) to the very task specific in that the tool queries for one specific diagnostic, such as inventorying installed software. The following section describes some of the low end interrogation tools that we examined.

One such tool, **Sysbotz**, is run as a shell script on Linux, and can provide a report on all software that is installed, the version numbers, summarize what software is missing, and report on all software that is not needed and/or that is outdated and needs to be upgraded. A simple query of a Linux system using commands at a prompt will return some of this information.

FreshDiagnose is a tool distributed as shareware on Windows operating systems. It returns complete and comprehensive diagnostics on the target system, produces reports, and saves the diagnostics for benchmarking performance statistics of various devices over time.

This tool interrogates a node's software system to return characteristics for, engines, environment information, file associations, libraries, memory, operating system, services, startup information, system files, and system policies. Diagnostics for the hardware system provides information on the BIOS, busses, cache memory, CMOS, memory, motherboard, port connectors, processor, and system slots. While this application cannot query remote systems, it seems to provide a good deal more information than some of the higher-end products and may be useful in assessing whether or not to apply a patch to a mission critical system.

Belarc Advisor is another free-for-personal-use tool that from Crucial Technology that scans a workstation and reports in HTML format the operating system, with Service Pack and Build number, all installed software including versions, license information, and where it is located on the drive, installed Microsoft hotfixes, and hard drive and processor information.

A manual inspection of the Task Manager in the Windows OS can provide information on services that are running at a given point in time on a PC, as well as CPU usage and memory usage. This type of information collected and tracked over time, in conjunction with other available inventory data, would be useful in determining the health of a system, and may aid in deciding whether or not a particular patch should be applied to a system.

While there are many packages available that will query networks and individual nodes for a variety of information, our search did not come up with a comprehensive application that could provide all of the desired attributes for determining the feasibility of applying a patch safely, with no adverse affects, to a mission critical system.

Such a package ideally would include software and hardware inventory information, data regarding the health of the system, benchmark information alerting to changes in the health of the system, such as a marked decrease in the amount of available hard drive space or the page file size, the node's role in the enterprise (a secondary backup system or a mission critical system), operating system information, etc. There appears to be a need for one application that can provide that information and make assessments regarding the feasibility of applying a patch to a particular system. Patch deployment based on those criteria is essential. Attributes that must be analyzed and assessed in order to develop a successful patch deployment policy.

2.4.3 Case Studies

In order to better illustrate the method and some of the critical decisions that users of such tools must consider, we developed case studies using several of the tools and technologies listed above. This study is not meant to be an endorsement of any of the tools and technologies utilized, rather our intent was to give the reader a better understanding of the process and critical decisions that need be made by the administrators. This process resulted in the following case study and observations.

Case Study 1: Sovereign Time

The Timestamp Server (TS) product from Sovereign Time is an appliance device that issues digital timestamps to protect the integrity of digital information. The TS system runs on a standard PC platform running Windows 2000 Workstation Operating System. The system is a security related device and is designed for deployment in an organization's Network Operation Centers (NOC), typically behind a firewall.

Beside the operating system, the TS application is built upon, and therefore dependant upon a variety of technologies, including:

- ✓ Tomcat Web Server
- ✓ IBM 4758 Cryptographic Coprocessor
- ✓ ACE Development Library
- ✓ Java SDK
- ✓ Xerces XML Parser

Application developers today have the advantage of leveraging technologies such as those stated above in order to develop more advanced applications and take full advantage of software reuse methodologies. In the case of the TS, these technologies significantly reduced the time it would take to develop all of its current capabilities from scratch. However, this makes patch and update management very difficult. Although the details of this report are centered on Windows® updates, it is interesting to consider the update/patch problem for the complete system.

The IBM 4758 Cryptographic Coprocessor lays the foundation for the TS platform. This processor hosts the primary timestamping component for the TS. The device is certified under FIPS 140-1 at levels 3 and 4 and it provides physical and logical security for the TS system. The IBM device uses a system of layers of security (or segments) to provide levels of protection for the device and for applications running inside of the device.

Segment 1, for example, provides a mini-boot capability that will load other software layers and enforce security policies within the device. Segments 2 and 3 are used for operating system and end user applications, respectively. In the development of the system, the design team made a decision to bind the segment 3 application to a specific revision of the Segment 1 system. The specific version that was selected was known to work with the application and targeted testing and empirical data backed up this decision. This binding gave confidence that no one can revert Segment 2 back to a version that may have some vulnerability. Future updates, however, present a problem for the application, as they are forced to update the application based on the release of updates for Segment 1 from IBM. This strict policy of controlling updates gives us, the vendor, control over exactly what versions of software will work together. Along with that level of control comes the responsibility to monitor, test, and distribute updates to the 4758 software and our application.

Vulnerability Considerations

The Windows 2000 operating system provides an easy platform for the TS system to be developed on and deployed on. All of the appropriate drivers and supporting technology are available. The TS application, however, uses very few native OS services. From a networking standpoint, the only ports that are open are handled either directly by the application or by the Tomcat web server.

In designing a strategy for applying patches, there are several considerations that must be made in determining how to test and deploy Microsoft patches. The considerations include security, process manageability, and customer confidence. Security is our foremost concern and the most obvious one. In our processes, ALL Microsoft patches are analyzed for security. Our security analysis considers how a patch directly affects our application as well as how it might indirectly affect the overall security of the device. In a recent example, a security patch was released for a security problem with Outlook Express. Although our TS application never uses the services of Outlook Express and the system is not intended for receiving mail of any sort, we still tested and deployed the patch. The reasoning is based on concerns that an attacker could still launch or exploit the Outlook Express vulnerability through some means that we cannot currently contemplate.

Customer confidence is another important consideration in our patching strategy. The TS system is sometimes deployed in larger organizations that have numerous Windows system or in organizations that are more concerned about security. Vulnerability scans and patch monitoring can reveal when the TS system is lagging behind in the update

process or when a specific update was missed or not deployed for some technical reasons. To avoid any questions or concerns in our customers, we analyze every update and strongly consider the ramifications of deployment. In our processes, most of the Microsoft patches are tested and recommended for deployment.

Overview of Overall Process

Microsoft releases patches the first Wednesday of each month and recommends that critical patches be applied within 24 hours. Our support department releases a report at the end of every month after testing has been performed. We then advise our customers on the advisability pf patching/not patching, as the case may be. Following is some of the patch information we acquire before deployment:

- ✓ Patch name
- ✓ Brief summary of patch
- ✓ Knowledge Base Article of Security Bulletin location
- ✓ Date published
- ✓ Date of reissue or re-release
- ✓ Affected Operating system or software
- ✓ Impact of Vulnerability (DDoS, etc.)
- ✓ Description
- ✓ Size of Patch
- ✓ Are there any mitigating factors?
- ✓ Affected files
- ✓ Deployment method
- ✓ Will the patch add, delete or replace files?

Testing Method

Testing Environment: The patches are applied using a 2-step process according to the critical status of the hardware on which they are applied. For stability purposes, it is imperative to verify that the installation of a patch does not bring down the device. Below is the list of the machines and the critical status of each machine.

Calibration #2 - not a production box, which has no critical impact if patching causes problems.

Cortland Ops 1, Cortland Ops 2, and Cortland TSS - production boxes, which have critical impact if brought out of commission.

Step 1 of the patching process involves installing the patch on the Calibration Unit to verify if there are any immediate ill effects of the installation. It is also important to know what to expect from the installation, whether or not it will require a reboot, what questions may be asked during the install, what visual changes may be expected and generally how the installer works. Step 2 of the patching process involves actually testing the functionality of the device. After the patch has been installed, basic functions are verified by performing audits, requesting timestamps, and walking through the web interface.

Recording test results: The test results are then recorded to show history, tests performed and results of the patching actions. The list below shows each item that is documented along with a description.

Patch Name - File name of the patch
Description - Patch verbose description
Test Period - Date patch was applied
Machines - Machines that the patch was applied to
Test Results - Detailed test results (reboot required? successful audit after reboot? forced audits from upper clock and from target machines successful?, scheduled audits successful?)

Releasing Customer Documentation: Once all testing is performed, documents must be formatted and released to customers showing the results of the tests. The next section shows the formats for the reports along with examples.

Report Formats

Reporting all actions performed is as important as doing the work. In-house technical data must be documented to capture as much information as possible for history purposes. However, customer correspondence does not demand the same. To solve this problem, there are 2 reports, each comprised of the data necessary for the personnel using the particular report.

Sovereign Time reports to our customers once a month advising them of the results of our patch deployment tests. As it happened during this period, 7 new Microsoft vulnerabilities were reported and our report to our customer was due within a week. Thus, we were unable to test deployment with our usual method: Fully test one patch at a time on the calibration machine and the machine run for a week before deploying to the operational timestamp servers. In this instance, all patches were deployed to the

calibration machine on a Friday. On Monday, it was determined that the calibration machine was successfully receiving audits and the patches were then deployed to the three operational timestamp servers.

The table on the following pages contains the information gathered about the latest round of Microsoft patches as described in Step 1 of our patching process.

All patches were considered critical. We deliberated about the installation of one of the patches, KB837009 which, through Microsoft Outlook, would allow remote code execution. Although Outlook is not used nor installed on the timestamping servers, if a remote scan of the target revealed its presence it is feasible that the machine would be vulnerable to attack if left unpatched. At some point, either intentionally or inadvertently, Outlook could be installed on the target machine.

However, since Outlook is not currently installed on the target machine and will not be in the foreseeable future, the addition/deletion/replacement of the dll files to the machine could interfere with the normal operation of other software crucial to the successful operation of the timestamp server should those happen to be shared files.

Due to the critical nature of the vulnerability and the absolute necessity to keep the timestamp servers operational at all times, we decided to test and deploy the Outlook Security Patch.

The examination of the files contained in the patch would, under normal circumstances, be outside of the scope of the average system administrator making a decision to patch or not. This is why it is imperative that a software solution be developed that would enable the average administrator to make a more informed decision, or rather, make that decision for the administrator.

Vendor Name	Patch Name	Operating System or Software	Date of Issue	Version Number or DOC Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Are there mitigating factors?	Affected Files	Deployment Method	Add, delete or replace files?	File Properties determinable Yes/No
Microsoft	KB831167	Windows 2000 SP4	April 09 2004	Knowledge Base Article 831167	Allows nonsecure connections to secure websites	Critical	An identified issue may cause errors when Internet Explorer attempts to renew a connection to a server. You should apply this update if you begin to receive errors connecting to websites after you have applied the KB832894 security update to Internet Explorer. After you install this item, you may need to restart your computer.	378 KB		Wininet.dll	Self-Extracting file	Replace	YES

Vendor Name	Patch Name	Operating System or Software	Date of Issue	Version Number or DOC Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Are there mitigating factors?	Affected Files	Deployment Method	Add, delete or replace files?	File Properties determinable Yes/No
Microsoft	KB837001	Windows 2000 SP4	April 12 2004	Security Bulletin MS04-014	Remote code Execution	Critical	A security issue has been identified that could allow an attacker to compromise a computer running Windows and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	2.8 MB		Dao360.dll Expsrv.dll Msexch40.dll Msexcl40.dll Msjet40.dll Msjetoledb40.dll Msjint40.dll Msjter40.dll Msjtes40.dll Msltus40.dll Mspbde40.dll Msrd2x40.dll Msrd3x40.dll Msrepl40.dll Mstext40.dll Mswdat10.dll Mswstr10.dll Msxbde40.dll Vbajet32.dll	Self-Extracting file	Replace	YES

Vendor Name	Patch Name	Operating System or Software	Date of Issue	Version Number or DOC Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Are there mitigating factors?	Affected Files	Deployment Method	Add, delete or replace files?	File Properties determinable Yes/No
Microsoft	KB828741	Windows 2000 SP4	April 09 2004	Security Bulletin MS04-012	Remote Code Execution	Critical	A security issue has been identified that could allow an attacker to compromise a computer running Windows and gain control over it. An attacker who successfully exploited the most severe of these vulnerabilities could take complete control of the affected system. An attacker could then take any action on the affected system, including installing programs; viewing, changing, or deleting data; or creating new accounts that have full privileges.	4.5 MB		Catsrv.dll Catsrvut.dll Clbcatex.dll Clbcatq.dll Colbact.dll Comadmin.dll Comrepl.dll Comsetup.dll Comsvcs.dll Comuid.dll Dtcsetup.exe Es.dll Msdtclog.dll Msdtcprx.dll Msdtctm.dll Msdtcui.dll Mtstocom.exe Mtxclu.dll Mtxdm.dll Mtxlegih.dll Mtxoci.dll Ole32.dll Rpcproxy.dll Rpcrt4.dll Rpcss.dll Txfaux.dll Xolehlp.dll	Self-Extracting file	Replacement	YES

Vendor Name	Patch Name	Operating System or Software	Date of Issue	Version Number or DOC Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Are there mitigating factors?	Affected Files	Deployment Method	Add, delete or replace files?	File Properties determinable Yes/No
Microsoft	KB835732	Windows 2000 SP4	April 09 2004	Security Bulletin MS04-011	Remote Code Execution	Critical	Multiple security issues have been identified that could allow an attacker to compromise a computer running Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	6.8 MB		Advapi32.dll Nmcom.dll Basesrv.dll Ntdll.dll Browser.dll Ntdsa.dll Callcont.dll Ntkrnlmp.exe Cmd.exe Ntkrnlpa.exe Crypt32.dll Ntkrpamp.exe Cryptnet.dll Ntoskrnl.exe Cryptsvc.dll Psbase.dll Dnsapi.dll Rdpwd.sys Dnsrslvr.dll Samlib.dll Eventlog.dll Samsrv.dll Gdi32.dll Scecli.dll H323.tsp Scesrv.dll Hfsecper.inf Schannel.dll Hfsecupd.inf Seclogon.dll Ipnathlp.dll Sfcfiles.dll Kdcsvc.dll Sp3res.dll	Self-Extracting file	Replacement	YES

Vendor Name	Patch Name	Operating System or Software	Date of Issue	Version Number or DOC Revision	Impact of Vulnerability	Severity Rating by Vendor	Description	Size of Patch	Are there mitigating factors?	Affected Files	Deployment Method	Add, delete or replace files?	File Properties determinable Yes/No
Microsoft	KB837009	Windows 2000 SP4	April 09 2004	Security Bulletin MS04-013	Remote Code Execution	Critical	A security issue has been identified in Microsoft Outlook Express that could allow an attacker to read files on your computer, or cause a program to run. You can help protect your computer by installing this update. After you install this item, you may have to restart your computer.	839 KB 5.5 1.9 MB - 6.0		Directdb.dll Inetcomm.dll Inetres.dll Msident.dll Msimn.exe Msoe.dll Msoeacct.dll Msoeres.dll Msoert2.dll Oeimport.dll Oemig50.exe Oemiglib.dll Wab.exe Wab32.dll Wabfind.dll Wabimp.dll Wabmig.exe	Self-Extracting file	Replacement	YES

After determining that these patches were appropriate to apply to timestamp servers, they were deployed as described above with the following results:

Patch Name	Test Period	Test Machine(s)	Test Results/Additional Information
Q831167bit64.exe	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(781 KB) failed to install ("Not valid Win 32 application")
Q837009v55.exe	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(839 KB) failed to install ("Requires Outlook Express 5.5 [SP2]")
Q837009v6.exe	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(855 KB) failed to install ("Requires Windows XP")
Win2000-KB828741-x86-ENU.EXE	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(4,568 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB835732-x86-ENU.EXE	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(6,993 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB837001-x86-ENU.EXE	04/23/04	Calibration #2 Product Version 3.3 Build Version 1.2	(2,903 KB) successfully installed, no reboot required after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Internet Explorer 6 SP1	04/26/04	Cortland-TSS Product Version 3.3 Build Version 1.1	Successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.

Patch Name	Test Period	Test Machine(s)	Test Results/Additional Information
Q831167bit32.exe	04/26/04	Cortland-TSS Product Version 3.3 Build Version 1.1	(378 KB) successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB828741-x86-ENU.EXE	04/26/04	Cortland-TSS Product Version 3.3 Build Version 1.1	(4,568 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB835732-x86-ENU.EXE	04/26/04	Cortland-TSS Product Version 3.3 Build Version 1.1	(6,993 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB837001-x86-ENU.EXE	04/26/04	Cortland-TSS Product Version 3.3 Build Version 1.1	(2,903 KB) successfully installed, no reboot required after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Internet Explorer 6 SP1	04/26/04	CORT-OP-1 Product Version 3.1 Build Version 2.5	Successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Q831167bit32.exe	04/26/04	CORT-OP-1 Product Version 3.1 Build Version 2.5	(378 KB) successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB828741-x86-ENU.EXE	04/26/04	CORT-OP-1 Product Version 3.1 Build Version 2.5	(4,568 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB835732-x86-ENU.EXE	04/26/04	CORT-OP-1 Product Version 3.1 Build Version 2.5	(6,993 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.

Patch Name	Test Period	Test Machine(s)	Test Results/Additional Information
Win2000-KB837001-x86-ENU.EXE	04/26/04	CORT-OP-1 Product Version 3.1 Build Version 2.5	(2,903 KB) successfully installed, no reboot required after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Internet Explorer 6 SP1	04/26/04	CORT-OP-2 Product Version 3.1 Build Version 2.5	Successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Q831167bit32.exe	04/26/04	CORT-OP-2 Product Version 3.1 Build Version 2.5	(378 KB) successfully installed, requested reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB828741-x86-ENU.EXE	04/26/04	CORT-OP-2 Product Version 3.1 Build Version 2.5	(4,568 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB835732-x86-ENU.EXE	04/26/04	CORT-OP-2 Product Version 3.1 Build Version 2.5	(6,993 KB) successfully installed, required reboot after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.
Win2000-KB837001-x86-ENU.EXE	04/26/04	CORT-OP-2 Product Version 3.1 Build Version 2.5	(2,903 KB) successfully installed, no reboot required after install. Received successful audit after rebooting as well as audit initiated from TSS and audit forced by upper clock. The following scheduled audits were successful.

We feel confident through our testing that the patches are critical to the security of our systems, that they do not adversely affect the operation of mission critical timestamp servers, and will advised our customers in the field that they are safe to apply to their systems.

Following is the report sent to our customer based on our findings.

Windows 2000 Patch Test Document

The following table lists information on the Microsoft Windows 2000 Service Packs and Critical security patches that have been applied to versions 3.1.2.5, 3.2.1.8, 3.3.1.1, 3.3.1.2 and 3.3.1.3 of the TSS/DAP. Operational and limited performance testing was performed after each patch and monitored for one week to verify there were no adverse affects caused by the upgrade.

Patch Information	Date Applied	Tests Results
Windows 2000 Service Pack 4 (SP4) provides the latest updates to the Windows 2000 family of operating systems in the following areas: security, operating system reliability, application compatibility, and setup. This service pack includes fully regression-tested versions of the patches for all security vulnerabilities affecting Windows 2000 found up to the closing date of Service Pack development. The following Microsoft Security Bulletins are included in Service Pack 4. MS03-025 (822679) - Flaw in Windows Message Handling through Utility Manager Could Enable Privilege Elevation MS03-024 (817606) - Buffer Overrun in Windows Could Lead to Data Corruption MS03-019 (817772) - Flaw in ISAPI extension for Windows Media Services could cause denial of service MS03-018 (811114) - Cumulative Patch for Internet Information Service MS03-014 (330994) - Cumulative Patch for Outlook Express MS03-013 (811493) - Buffer Overrun in Windows Kernel Message Handling Could Lead to Elevated Privileges MS03-010 (331953) - Flaw in RPC Endpoint Mapper Could Allow	12/05/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.

Patch Information	Date Applied	Tests Results
Denial of Service Attacks MS03-008 (814078) - Flaw in Windows Script Engine Could Allow Code Execution MS03-007 (815021) - Unchecked buffer in Windows Component Could Cause Web Server Compromise MS03-001 (810833) - Unchecked Buffer in Locator Service Could Lead to Code Execution MS02-071 (328310) - Flaw in Windows WM_TIMER Message Handling Could Enable Privilege Elevation MS02-070 (329170) - Flaw in SMB Signing Could Enable Group Policy to be Modified MS02-065 (329414) - Buffer Overrun in Microsoft Data Access Components Could Lead to Code Execution MS02-063 (329834) - Unchecked Buffer in PPTP Implementation Could Enable Denial of Service Attacks MS02-055 (323255) - Unchecked Buffer in Windows Help Facility Could Enable Code Execution MS02-053 (324096) - Buffer Overrun in SmartHTML Interpreter Could Allow Code Execution MS02-051 (324380) - Cryptographic Flaw in RDP Protocol can Lead to Information Disclosure MS02-050 (329115) - Certificate Validation Flaw Could Enable Identity Spoofing MS02-048 (323172) - Flaw in Certificate Enrollment Control Could Allow Deletion of Digital Certificates MS02-045 (326830) - Unchecked Buffer in Network Share Provider can Lead to Denial of Service MS02-042 (326886) - Flaw in Network Connection Manager Could Enable Privilege Elevation MS02-032 (320920) - Cumulative Patch for Windows Media Player MS01-022 (296441) - WebDAV Service Provider Can Allow Scripts to		

Patch Information	Date Applied	Tests Results
Levy Requests as User		
Post SP4 Critical Security Patches		
823559: Security Update for Microsoft Windows - (Posted Date: November 14, 2003) Download size: **382 KB** An identified security issue in Microsoft Windows could allow an attacker to compromise a Microsoft Windows-based system and then take a variety of actions. For example, an attacker could execute code on the system. By installing this update, you can help protect your computer. After you install this item, you may have to restart your computer.	11/19/03	Reboot was not required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Windows 2000 (KB824146) - (Posted Date: September 11, 2003) Download size: **917 KB** A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft® Windows® and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	11/25/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.

Patch Information	Date Applied	Tests Results
Security Update for Microsoft Windows (KB824141) - (Posted Date: October 17, 2003) Download size: **3.4 MB** A security issue has been identified that could allow an attacker to compromise a computer running Microsoft Windows and gain control over it. To attempt an attack, the attacker would have to be able to log on to the computer. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	12/08/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.

Patch Information	Date Applied	Tests Results
Security Update for Windows 2000 (KB823182) - (Posted Date: November 17, 2003) Download size: **359 KB** A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft Windows and gain complete control over it. For example, an attacker could execute code on your system. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	11/20/03	Reboot was not required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Windows (KB824105) - (Posted Date: September 09, 2003) Download size: **321 KB** A security issue has been identified in Microsoft Windows that could allow an attacker to see information in your computer's memory over a network. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	11/21/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Windows 2000 (KB826232) - (Posted Date: October 29, 2003) Download size: **329 KB** A security issue has been identified that could allow an attacker to read files or run programs on a computer, running Microsoft® Windows® 2000, that has been used to view an attacker's Web site or has read a specially crafted HTML e-mail. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	12/02/03	Reboot was not required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Windows 2000 (KB825119) - (Posted Date: October 13, 2003) Download size: **304 KB** A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft® Windows® 2000 and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	12/08/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced

Patch Information	Date Applied	Tests Results
		audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Windows 2000 (KB828035) - (Posted Date: October 29, 2003) Download size: **343 KB** A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft® Windows® 2000 and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	12/04/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Windows (KB828749) - (Posted Date: November 06, 2003) Download size: **329 KB** A security issue has been identified that could allow an attacker to remotely compromise a computer running Microsoft Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	12/08/03	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Microsoft Data Access Components (KB832483) - (Posted Date: January 13, 2004) Download size: **2 MB** An identified security issue in Microsoft Data Access Components could allow an attacker to compromise a Windows-based system and take a variety of actions. For example, an attacker could execute code on the system. By installing this update, you help protect your computer. After you install this item, you may have to restart your computer. Once you have installed this item, it cannot be removed.	1/13/04	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Windows 2000 (KB828028) - (Posted Date: February 09, 2004) Download size: **309 KB** A security issue has been identified in Microsoft Windows-based	02/19/04	Reboot required after installation of patch

Patch Information	Date Applied	Tests Results
systems that could allow an attacker to compromise your Microsoft Windows-based system and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may need to restart your computer.		After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Cumulative Security Update for Internet Explorer 6 Service Pack 1 (KB832894) - (Posted Date: January 30, 2004) Download size: **2.8 MB** Identified security issues in Internet Explorer could allow an attacker to compromise a Windows-based system. For example, an attacker could run programs on your computer while you view a Web page. This affects all computers with Internet Explorer installed (even if you don't run Internet Explorer as your Web browser). After you install this item, you may need to restart your computer.	02/19/04	Reboot required after installation of patch After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Critical Update for Internet Explorer 6 Service Pack 1 (KB831167) - (Posted Date: April 09, 2004) Download size: **378 KB** An identified issue may cause errors when Internet Explorer attempts to renew a connection to a server. You should apply this update if you begin to receive errors connecting to websites after you have applied the KB832894 security update to Internet Explorer. After you install this item, you may need to restart your computer. Comes in 32-bit and 64-bit	04/23/04	Utilized the 32-bit version. Successfully installed (requested reboot after install) After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Windows 2000 (KB837001) - (Posted Date: April 12, 2004) Download size: **2.8 MB** A security issue has been identified that could allow an attacker to compromise a computer running Windows and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	04/23/04	successfully installed (No reboot required after install) After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.

Patch Information	Date Applied	Tests Results
Security Update for Windows 2000 (KB828741) - (Posted Date: April 09, 2004) Download size: **4.5 MB** A security issue has been identified that could allow an attacker to compromise a computer running Windows and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	04/23/04	Successfully installed (required reboot after install) After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Security Update for Windows 2000 (KB835732) - (Posted Date: April 09, 2004) Download size: **6.8 MB** Multiple security issues have been identified that could allow an attacker to compromise a computer running Windows and gain complete control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.	04/23/04	Successfully installed (required reboot after install) After reboot, TSS was warranted and continued to work as specified. All forced audits and scheduled audits worked as required. Timestamping performance and operation were not affected.
Cumulative Security Update for Outlook Express 6 Service Pack 1 (KB837009) - (Posted Date: April 09, 2004) Download size: **1.9 MB** A security issue has been identified in Microsoft Outlook Express that could allow an attacker to read files on your computer, or cause a program to run. You can help protect your computer by installing this update. After you install this item, you may have to restart your computer	Pending	**This patch is distributed in 2 versions, Outlook Express 5.5 SP2, and Outlook Express 6 SP1. Currently waiting on installation of SP2 for Outlook Express 5.5 before installation of this patch**

Case Study 2: Automated Patch Management Systems

This was a test implementing a patch with an automated patch management system on one of our operational, Microsoft Professional client machines with a Microsoft patch that was missing from that client.

Before patching, we wanted to be sure of the health of the client as a precaution. Implementing a patch across the network could present special problems if the target is malfunctioning in some way in terms of networking components, hard disk space, memory problems, etc. We also wanted to ensure to appropriateness of the patch, and the importance of the patch.

In order to prepare the test environment, we wanted to:

- ✓ Determine what patch management system to use, and install it on a server
- ✓ Install any necessary patch agent pieces that were needed on the target/client machine
- ✓ Using the patch management software, choose a patch to apply to the client
- ✓ Install a tool on the client to determine the overall health of the client machine
- ✓ Install a tool on the client to determine the appropriateness of the patch for the client (software interrogation)
- ✓ Install a tool on the client to provide benchmarking information about the target to further verify the client's durability in accepting a patch
- ✓ Once we were satisfied that the client was in a state for accepting a patch, we analyzed the patch in order to consider the affects it may have on our client once patched, e.g. does this application require a reboot, if the machine is rendered inoperable, how will it affect the continuity of service to our customers

For this simple patching operation, we expected the process would be completed within a matter of minutes. We chose PatchLink Update 5.0, an automated, cross-platform security patch management system from PatchLink Corporation.

For the purposes of our test, we chose to review the MS04-014 837001 patch from Microsoft. There is a vulnerability in the Microsoft Jet Database Engine, which is in all installations of Microsoft Professional. According to Microsoft, "A buffer overrun vulnerability exists in the Microsoft Jet Database Engine (Jet) that could allow remote code execution. An attacker who successfully exploited this vulnerability could take complete control of an affected system, including installing programs; viewing, changing, or deleting data; or creating new accounts that have full privileges." Microsoft rates this patch "Critical".

To perform the test, PatchLink Update 5.0 was applied to on a newly installed Microsoft Windows Server machine, named WETSTONETEST, that was put on our inside network. The application required that IIS be on the administration machine (server machine).

For the client machine, we used an operational Microsoft Windows Professional machine, named WTSTN-SAW2, used for administration of our Seeing Stone Managed Security Services. This client machine is simply configured with tools necessary to connect to and administer our various sensors in the field. This machine is also configured with the Microsoft Outlook e-mail account, Seeing Stone Operations, and has a history of all e-mails sent to and from the Operation Center to all of our Seeing Stone customers.

Should the client machine be disconnected due to rebooting makes no difference to the operations of our Seeing Stone Service. The functions provided by this machine can be performed on any available Windows Professional or Server machine. It is a matter of convenience to the staff who administers the Seeing Stone Services to use this machine as it contains tools such as NetScan Tools, has pertinent web pages bookmarked for research, and contains the Seeing Stone Operations Outlook mailbox.

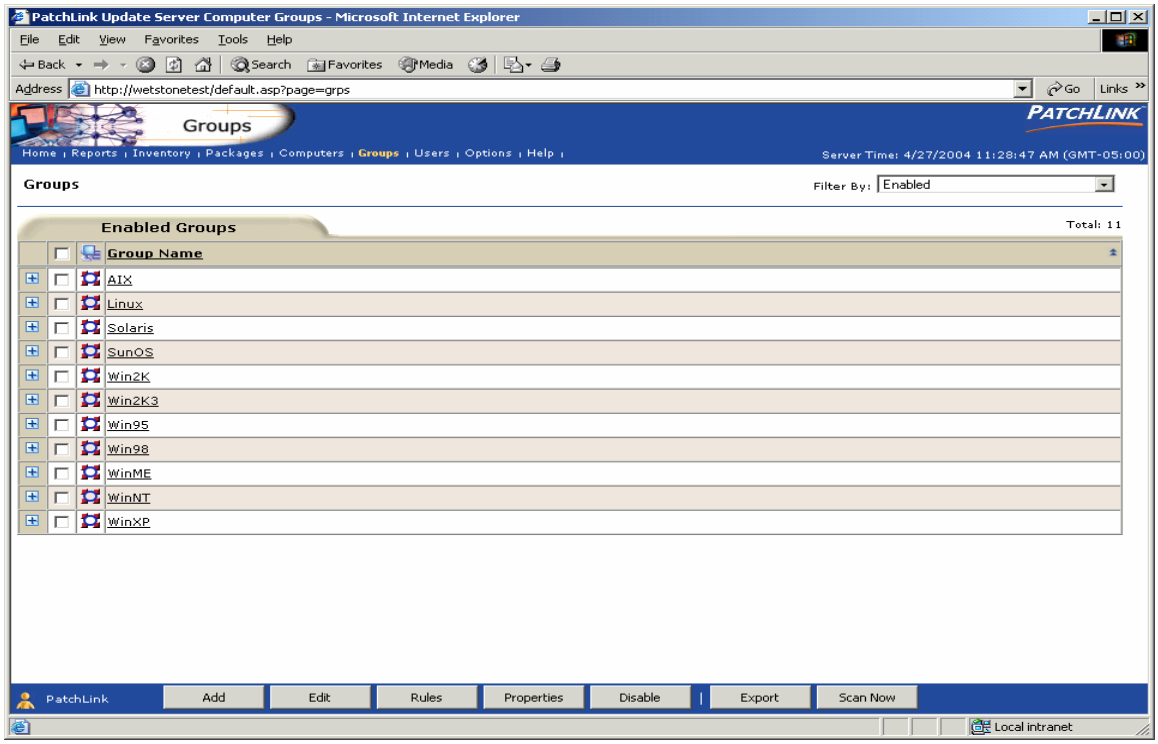
During this case study, we used BCM Health Monitor to monitor the health of the target system, and BCM Diagnostics to give us information about the condition of the target's processor, hard disk, and memory. We also used this tool to perform an overall stress test of the client machine.

We used Belarc Advisor to interrogate the client for operating system information and to verify that the patch in question had not already been applied.

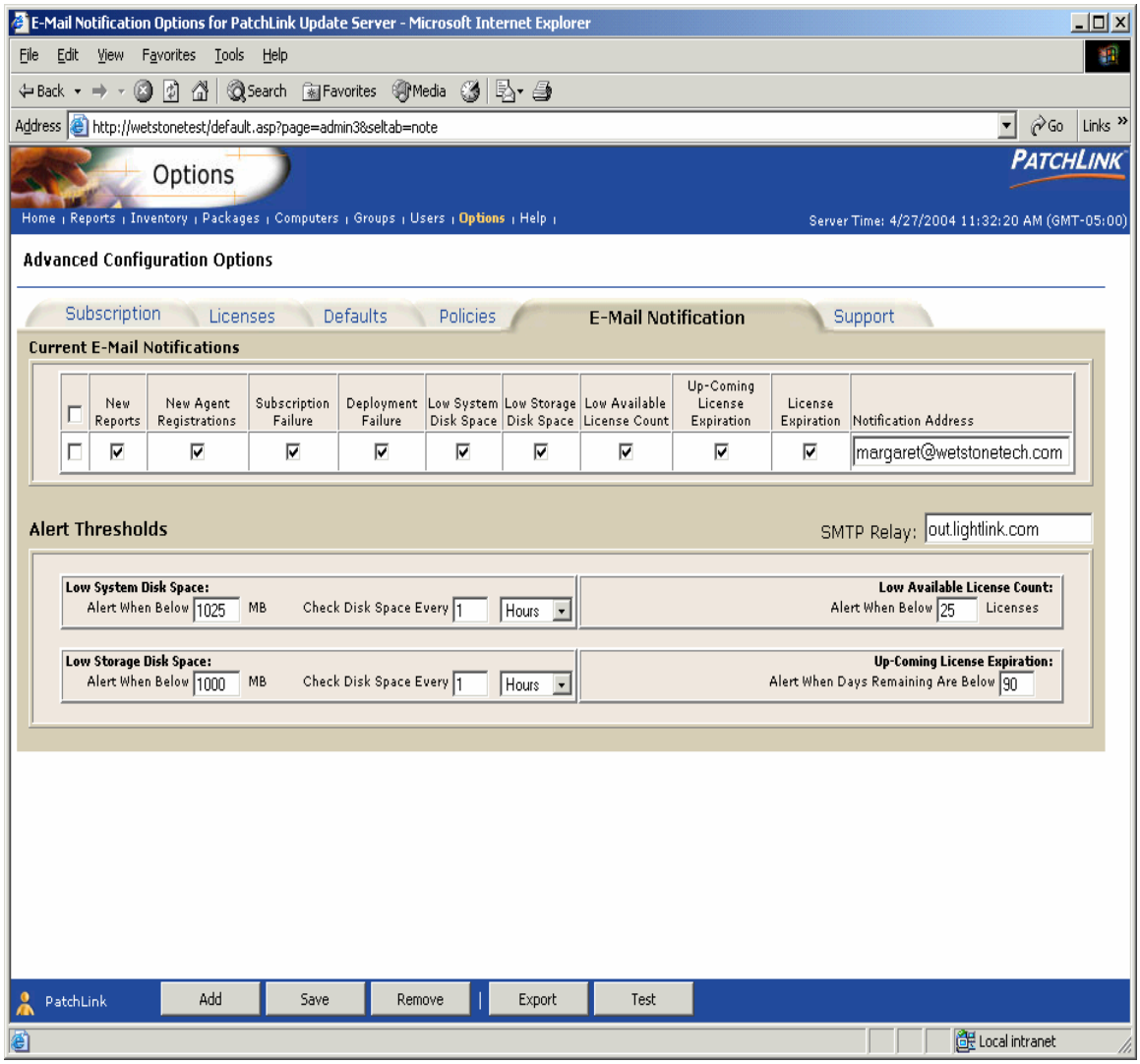
Finally FreshDiagnose 6.60 was installed on the client to give us benchmarking information on the network across which the patch would be applied, in addition to benchmarking information on the hard disk. This helps us further determine the health condition of the client before implementing the chosen patch.

PatchLink Update Agent was installed on the client (patch target) machine. This piece was necessary for the Update Server to communicate with the client over the network.

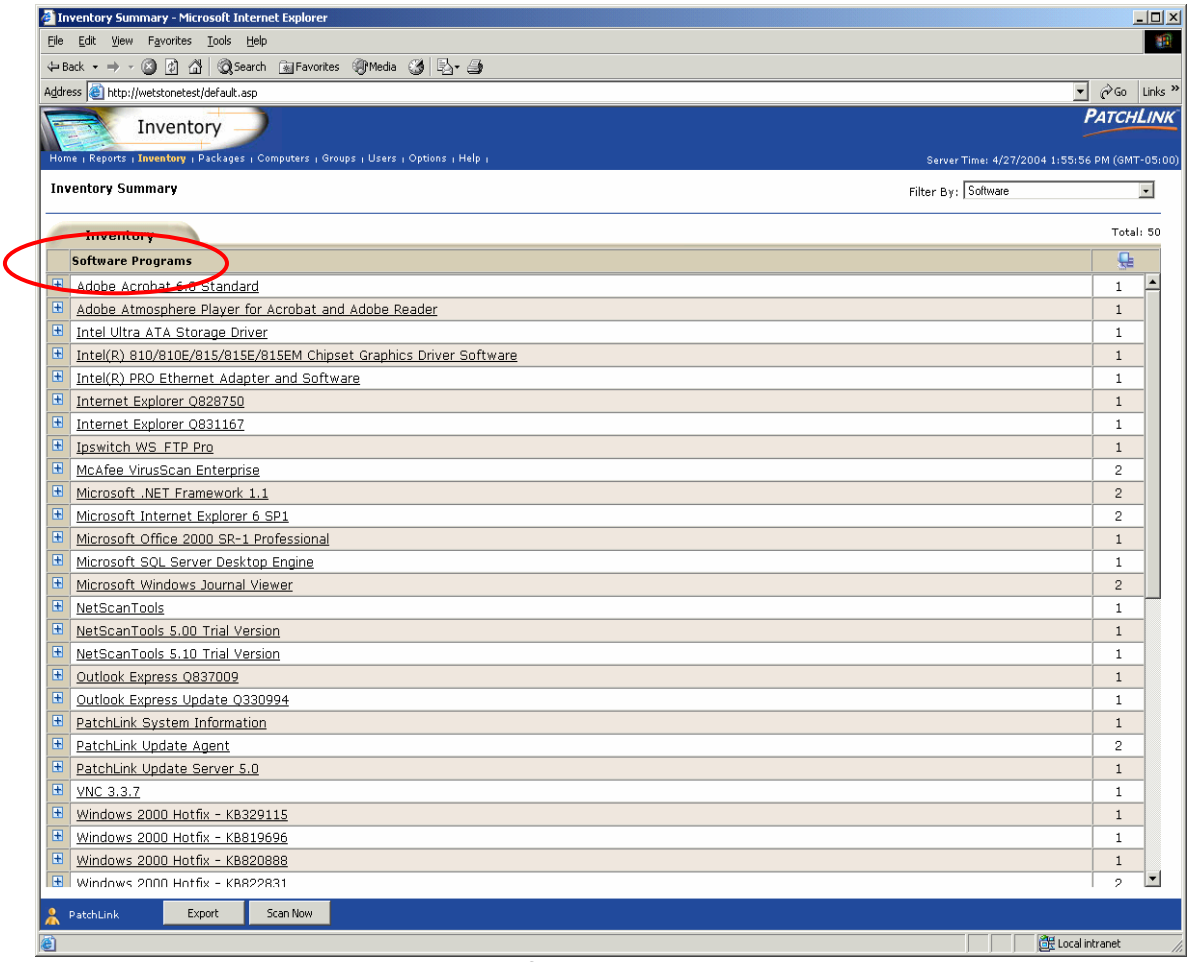
The screenshot below shows the "groups" that can be enabled to be patched across the network form a server machine.



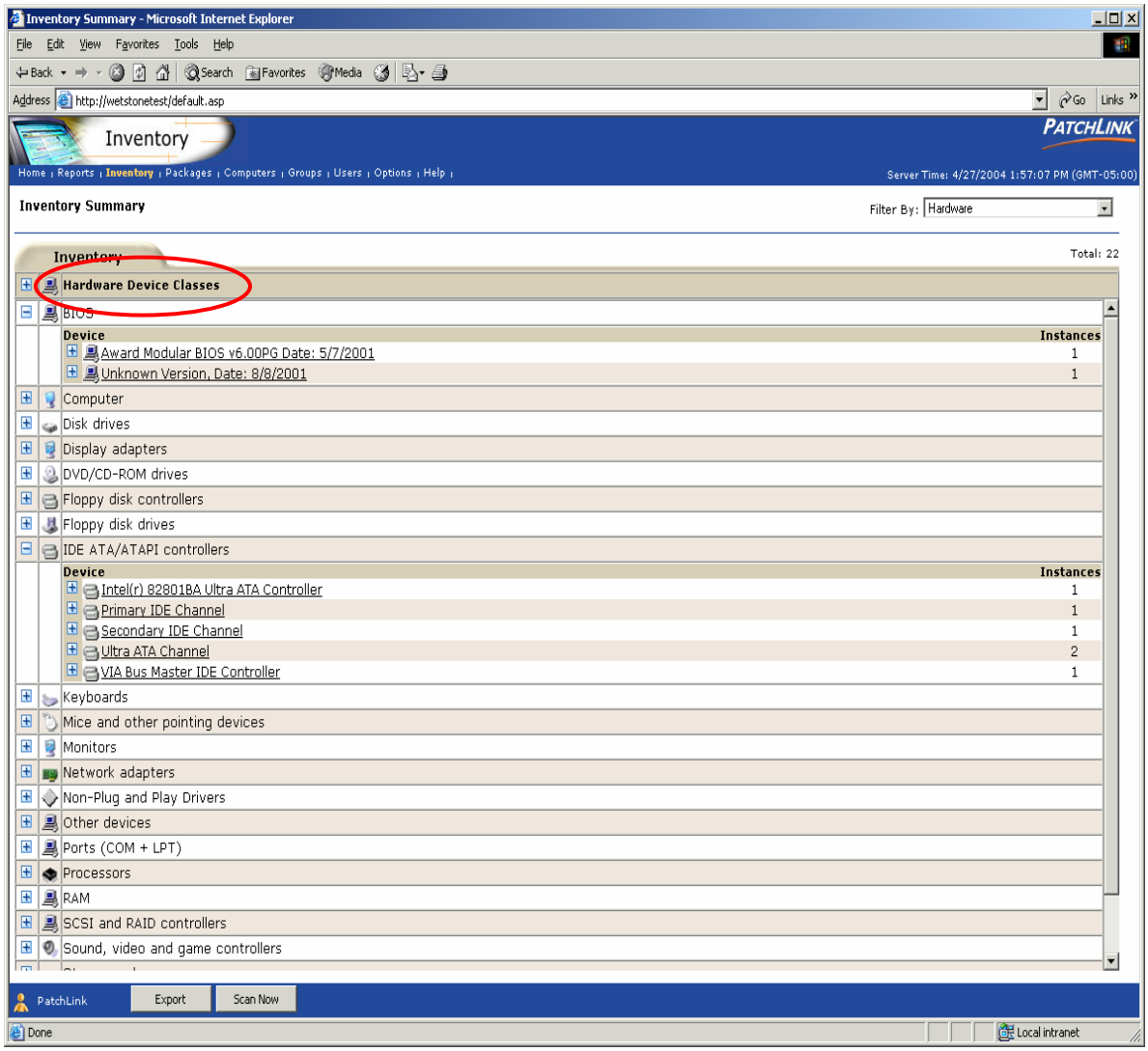
The software is extremely high end. Among the many tasks it can perform is the ability to e-mail patch results to whomever you decide.



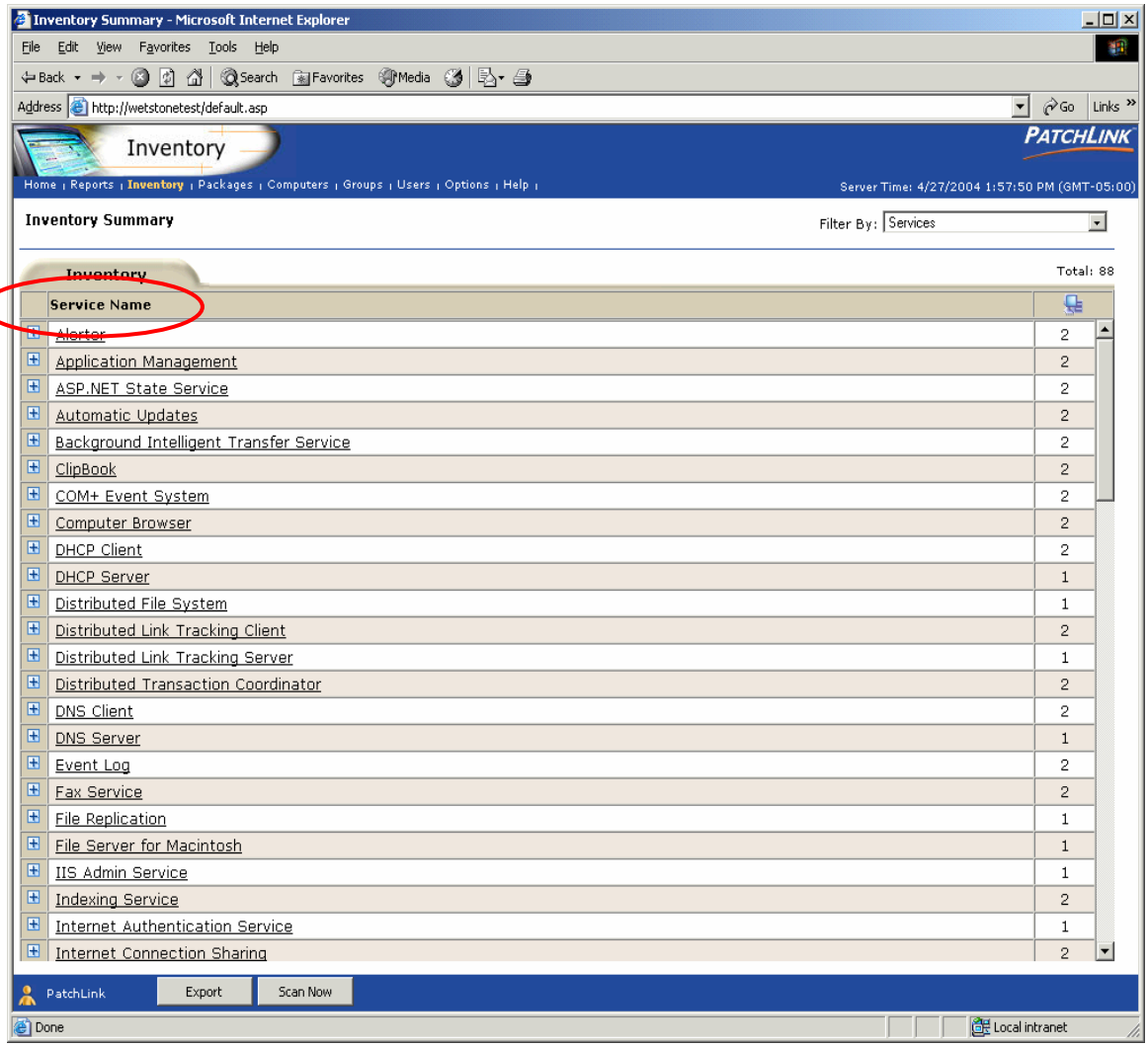
While PatchLink Update does an inventory of the target machine (see example screenshots below) it did not perform general health interrogation of the target machine.



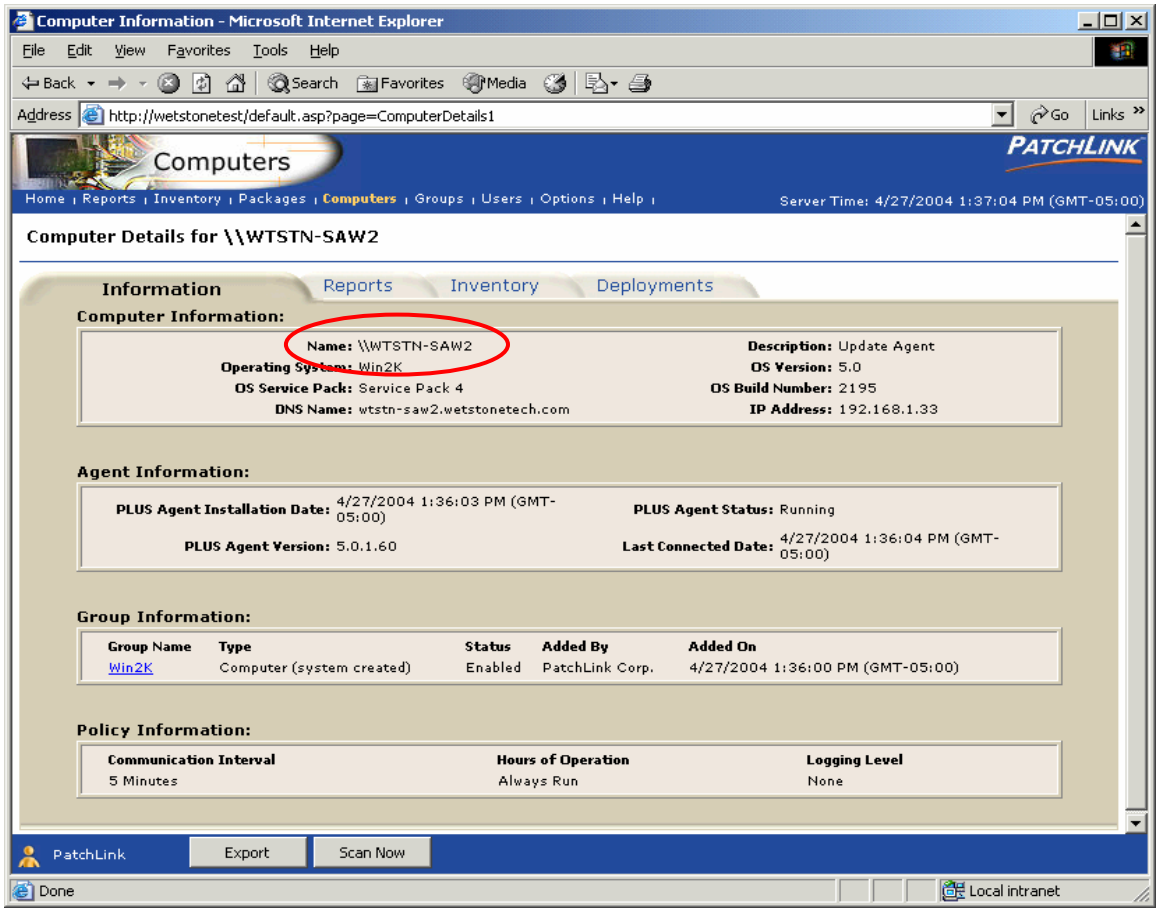
Software Inventory



Hardware Device Inventory



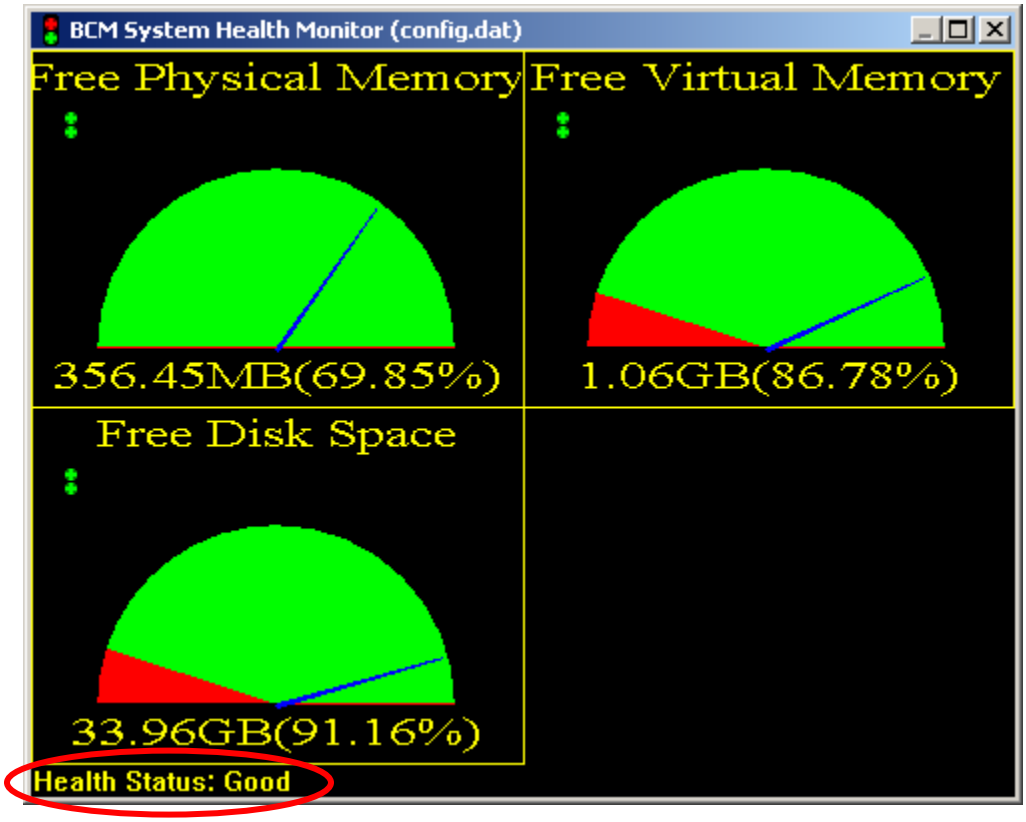
Services Inventory



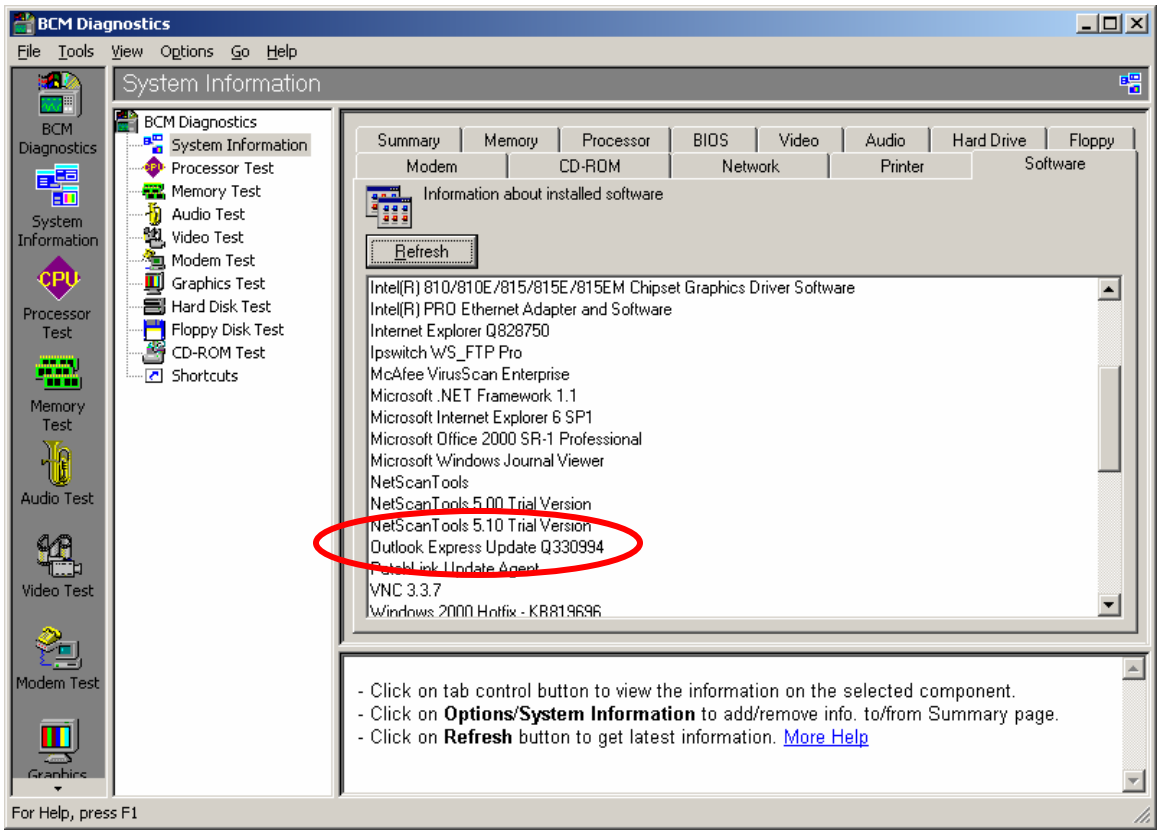
General Information

In order to obtain additional information about the client machine, and to verify the information received from PatchLink Update, three additional pieces of software were installed on the target machine.

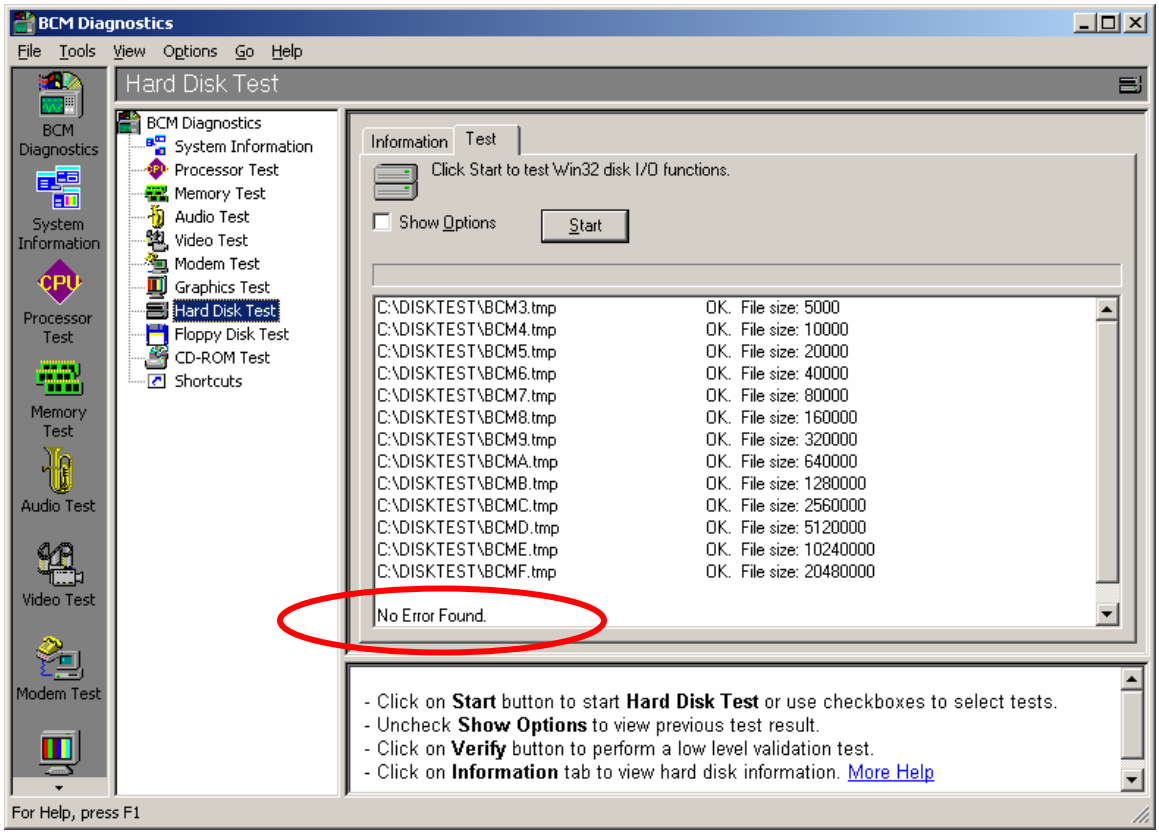
BCM Advanced ToolBox, BCM Advanced Research, Inc. was installed to provide answers to general health questions about the target machine. This software also gave us information about the health of various installed components of the machine using its stress test feature.



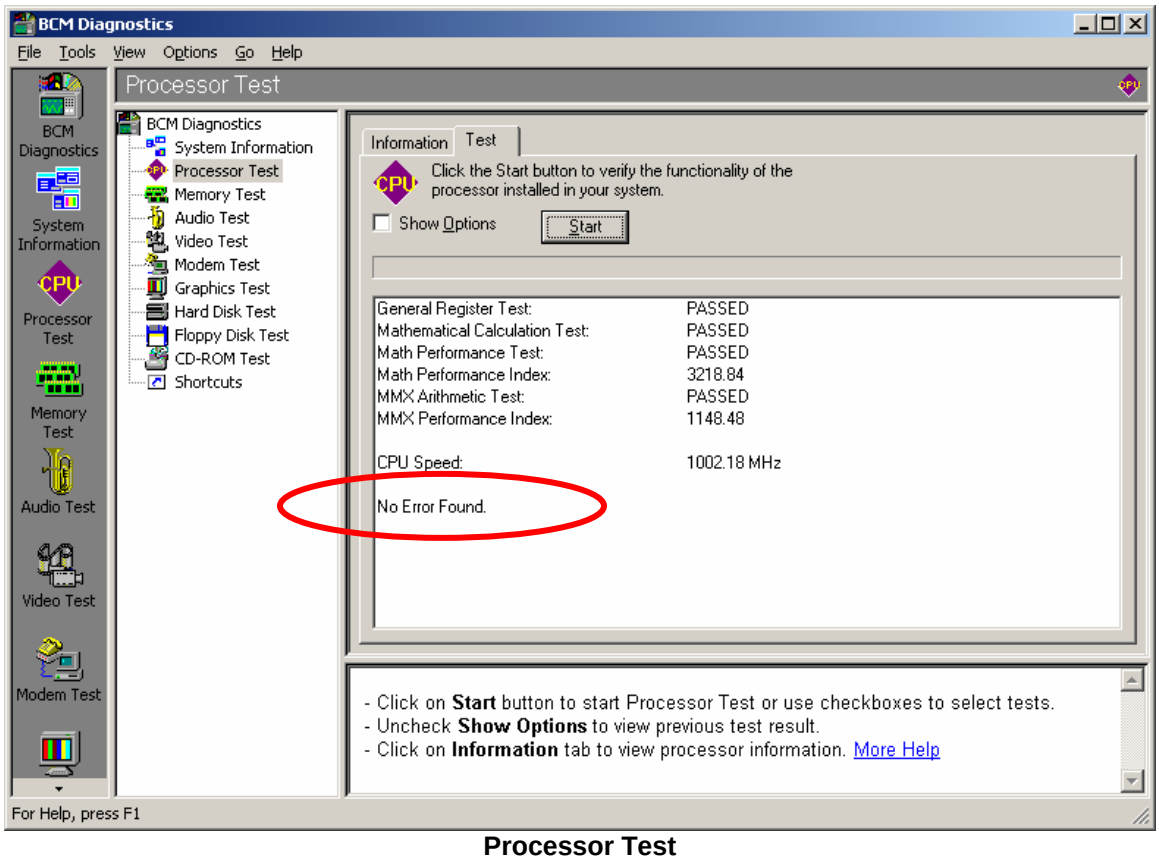
BCM Health Monitor



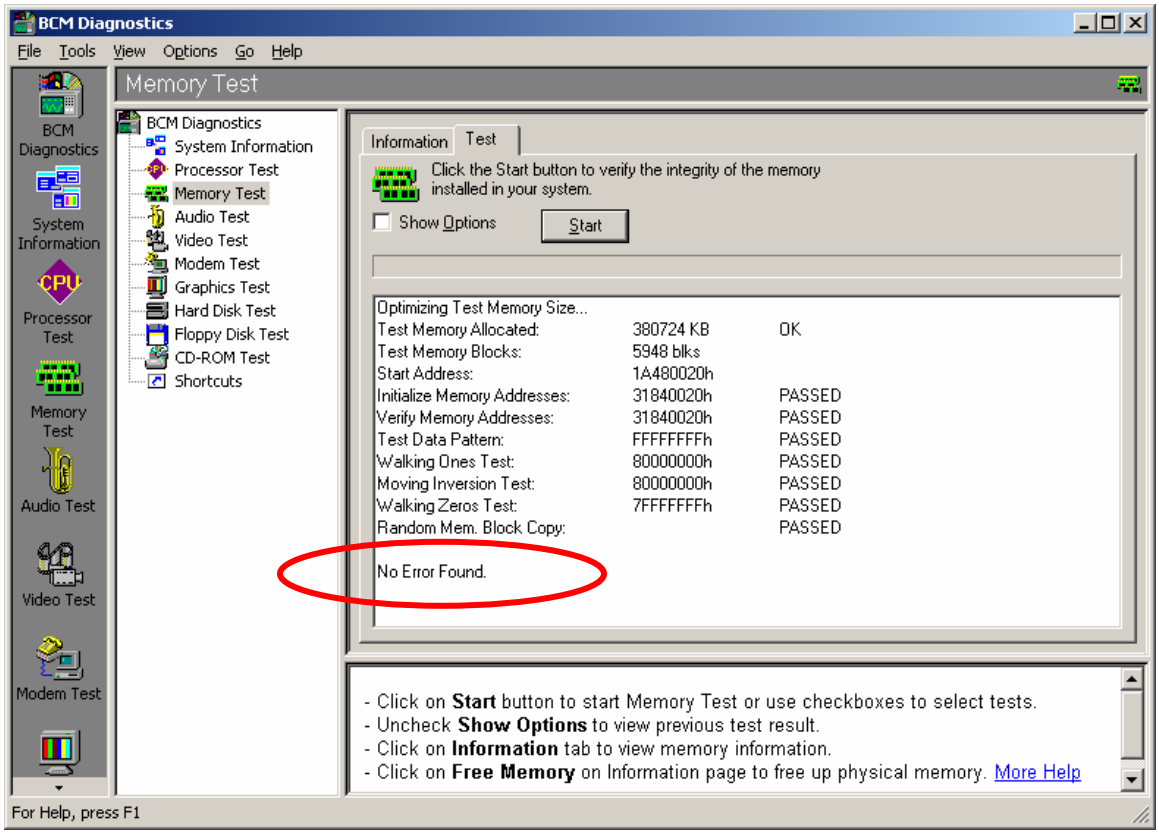
System Information – Including Patches Applied



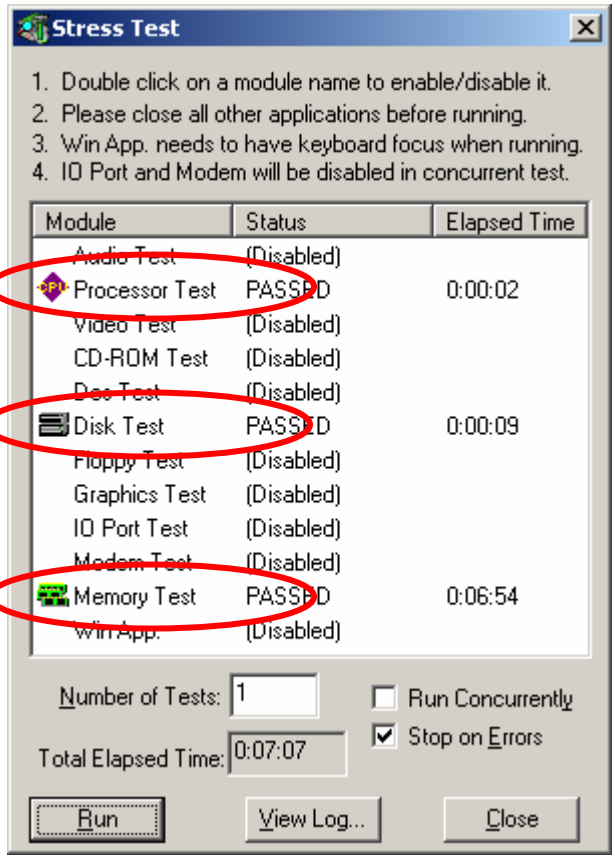
Test of Hard Disk

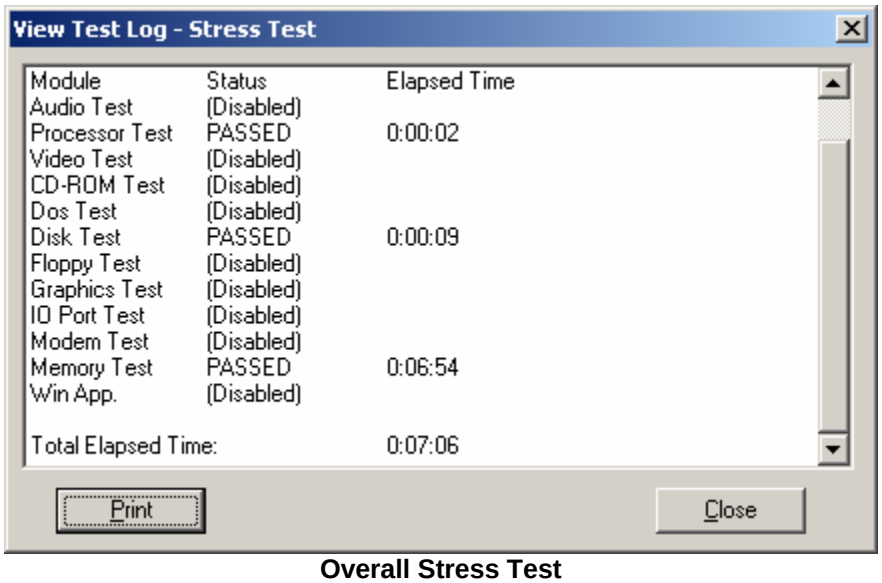


Processor Test


BCM Diagnostics
File Tools View Options Go Help
Memory Test
BCM Diagnostics
System Information
Processor Test
Memory Test
Audio Test
Video Test
Modem Test
Graphics Test
Hard Disk Test
Floppy Disk Test
CD-ROM Test
Shortcuts
Information Test
Click the Start button to verify the integrity of the memory installed in your system.
Show Options
Start
Optimizing Test Memory Size...
Test Memory Allocated: 380724 KB OK
Test Memory Blocks: 5948 blks
Start Address: 1A480020h
Initialize Memory Addresses: 31840020h PASSED
Verify Memory Addresses: 31840020h PASSED
Test Data Pattern: FFFFFFFFh PASSED
Walking Ones Test: 80000000h PASSED
Moving Inversion Test: 80000000h PASSED
Walking Zeros Test: 7FFFFFFFh PASSED
Random Mem. Block Copy: PASSED
No Error Found.
- Click on Start button to start Memory Test or use checkboxes to select tests.
- Uncheck Show Options to view previous test result.
- Click on Information tab to view memory information.
- Click on Free Memory on Information page to free up physical memory. More Help
For Help, press F1

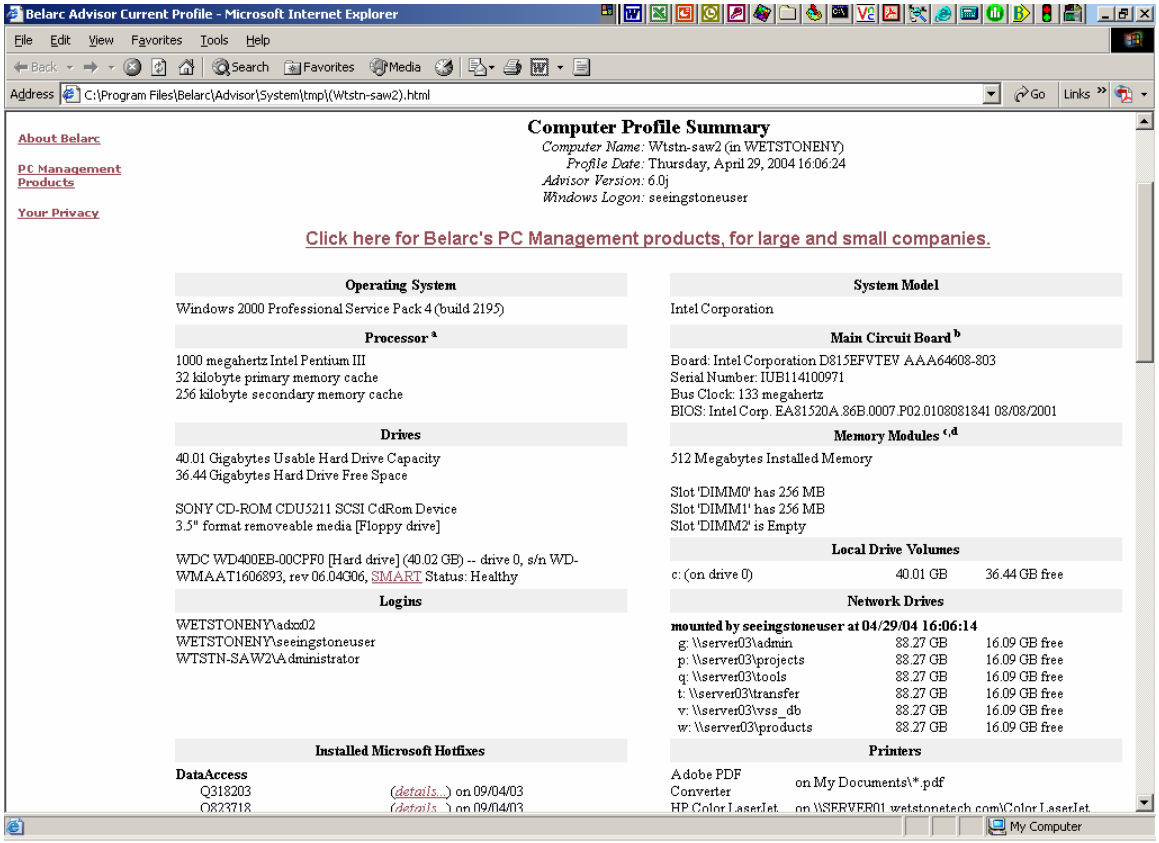
Memory Test

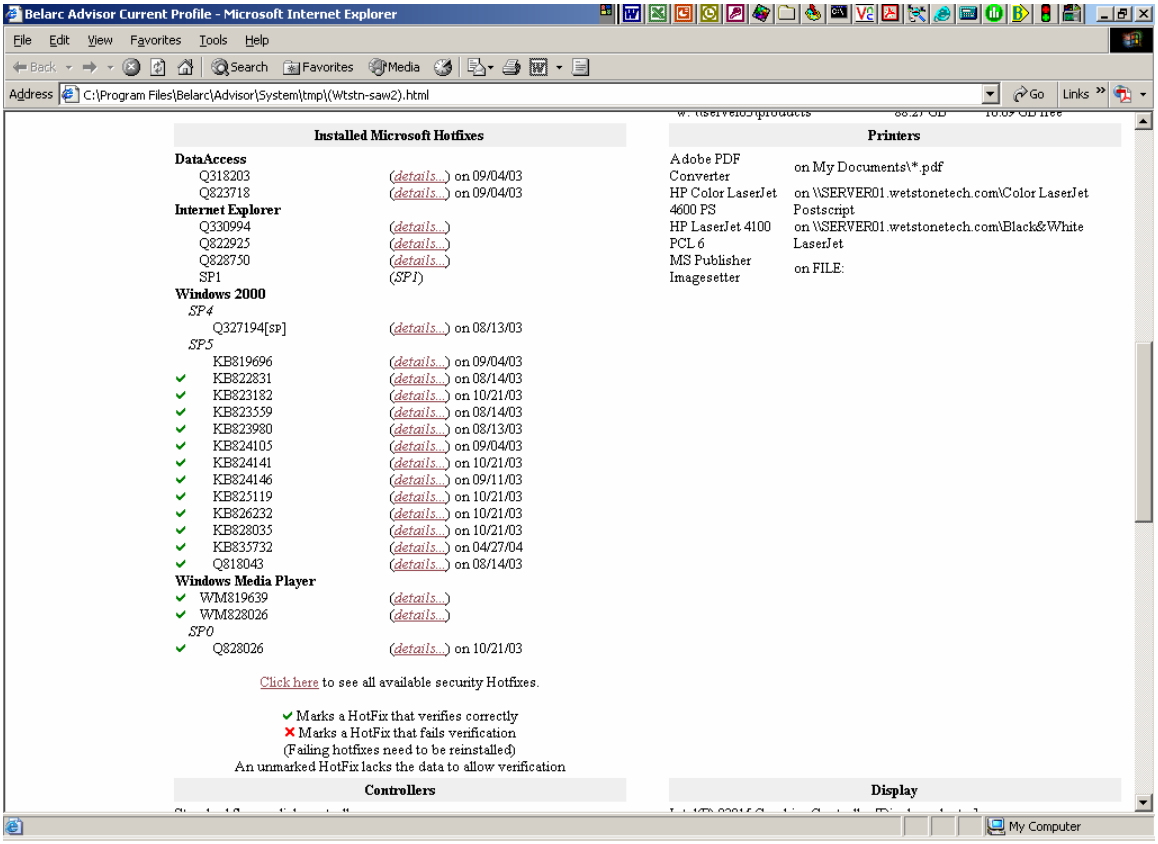


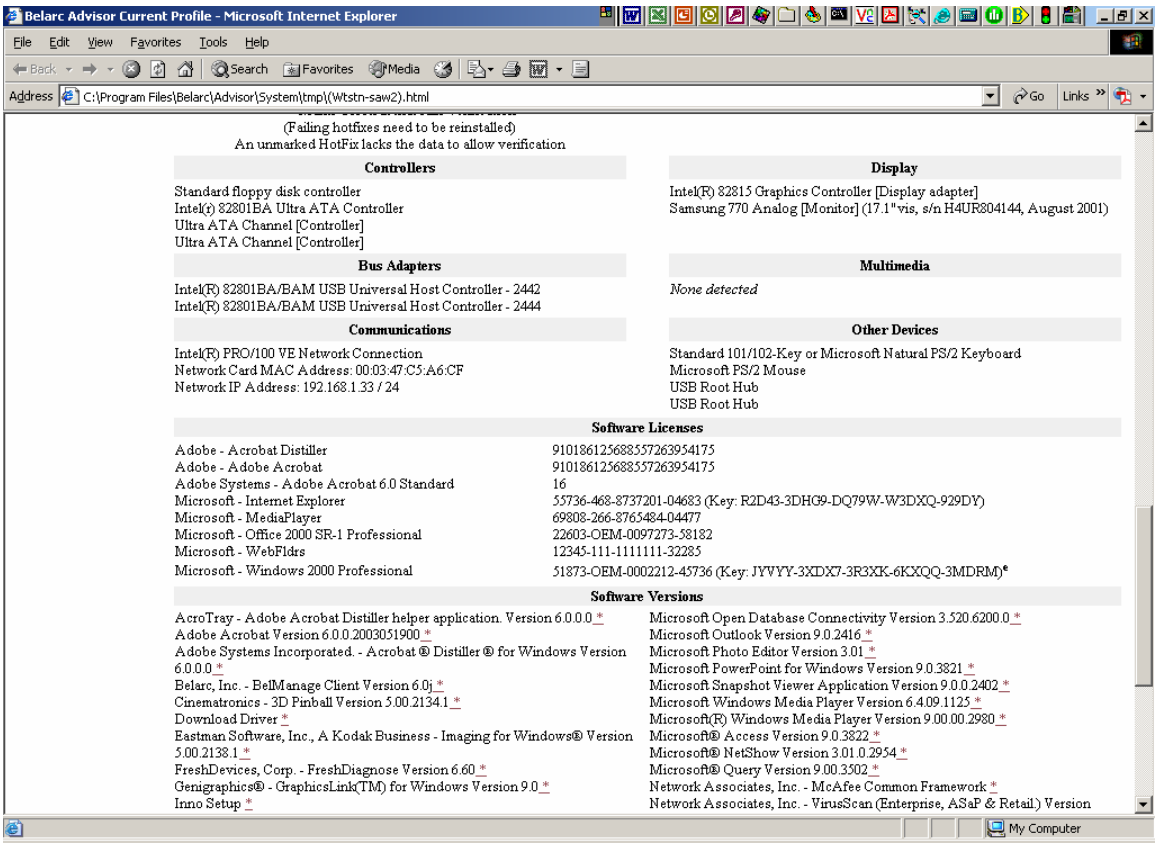


Overall Stress Test

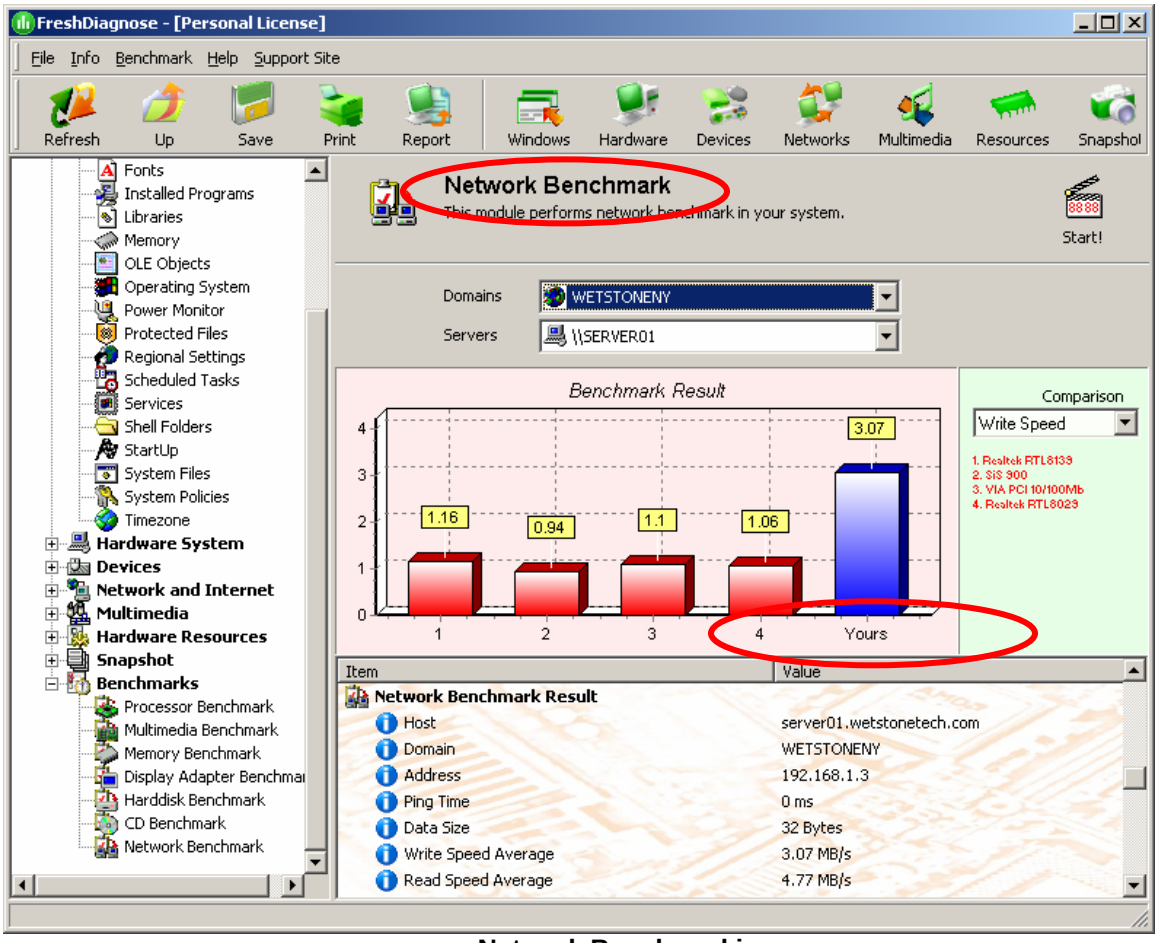
Belarc Advisor, Belarc, Inc. is another lower-end software package that was installed on the target machine. It opens in a single HTML screen that provides some of the information that can be obtained through PatchLink Update, such as the software inventory, patches applied, memory information, drive information, etc.



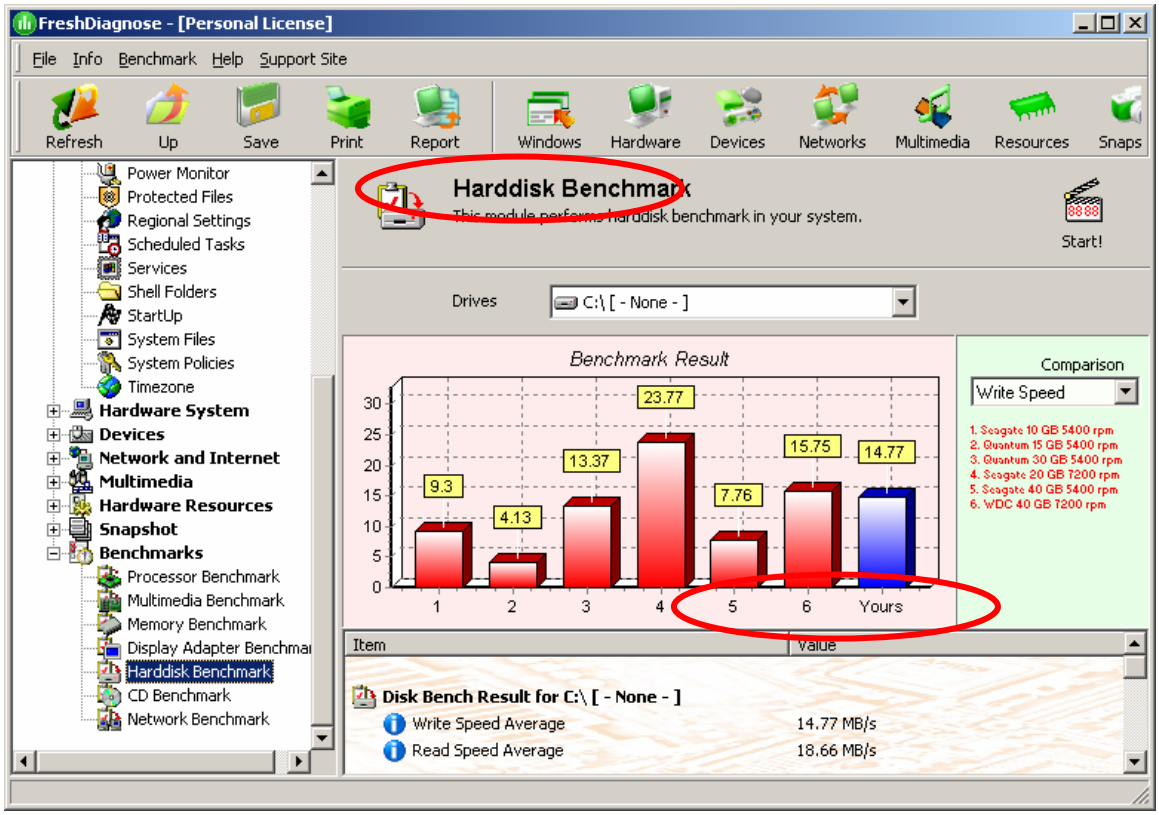

Belarc Advisor Current Profile - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Program Files\Belarc\Advisor\System\tmp\(Wtstn-saw2).html
Go Links
Installed Microsoft Hotfixes
DataAccess
Q318203 (details...) on 09/04/03
Q823718 (details...) on 09/04/03
Internet Explorer
Q330994 (details...)
Q822925 (details...)
Q828750 (details...)
SP1 (SP1)
Windows 2000
SP4
Q327194[SP] (details...) on 08/13/03
SP5
KB819696 (details...) on 09/04/03
KB822831 (details...) on 08/14/03
KB823182 (details...) on 10/21/03
KB823559 (details...) on 08/14/03
KB823980 (details...) on 08/13/03
KB824105 (details...) on 09/04/03
KB824141 (details...) on 10/21/03
KB824146 (details...) on 09/11/03
KB825119 (details...) on 10/21/03
KB826232 (details...) on 10/21/03
KB828035 (details...) on 10/21/03
KB835732 (details...) on 04/27/04
Q818043 (details...) on 08/14/03
Windows Media Player
WM819639 (details...)
WM828026 (details...)
SP0
Q828026 (details...) on 10/21/03
Click here to see all available security Hotfixes.
Marks a HotFix that verifies correctly
Marks a HotFix that fails verification
(Failing hotfixes need to be reinstalled)
An unmarked HotFix lacks the data to allow verification
Printers
Adobe PDF Converter on My Documents*.pdf
HP Color LaserJet 4600 PS on \\SERVER01.wetstonetech.com\Color LaserJet Postscript
HP LaserJet 4100 PCL 6 on \\SERVER01.wetstonetech.com\Black&White LaserJet
MS Publisher Imagesetter on FILE:
Controllers
Display
My Computer



Lastly, we installed FreshDiagnose 6.60 from Freshdevices Corp. This application provided many of the interrogation results provided by other products, in addition to benchmark information. Below are a few benchmarking examples.

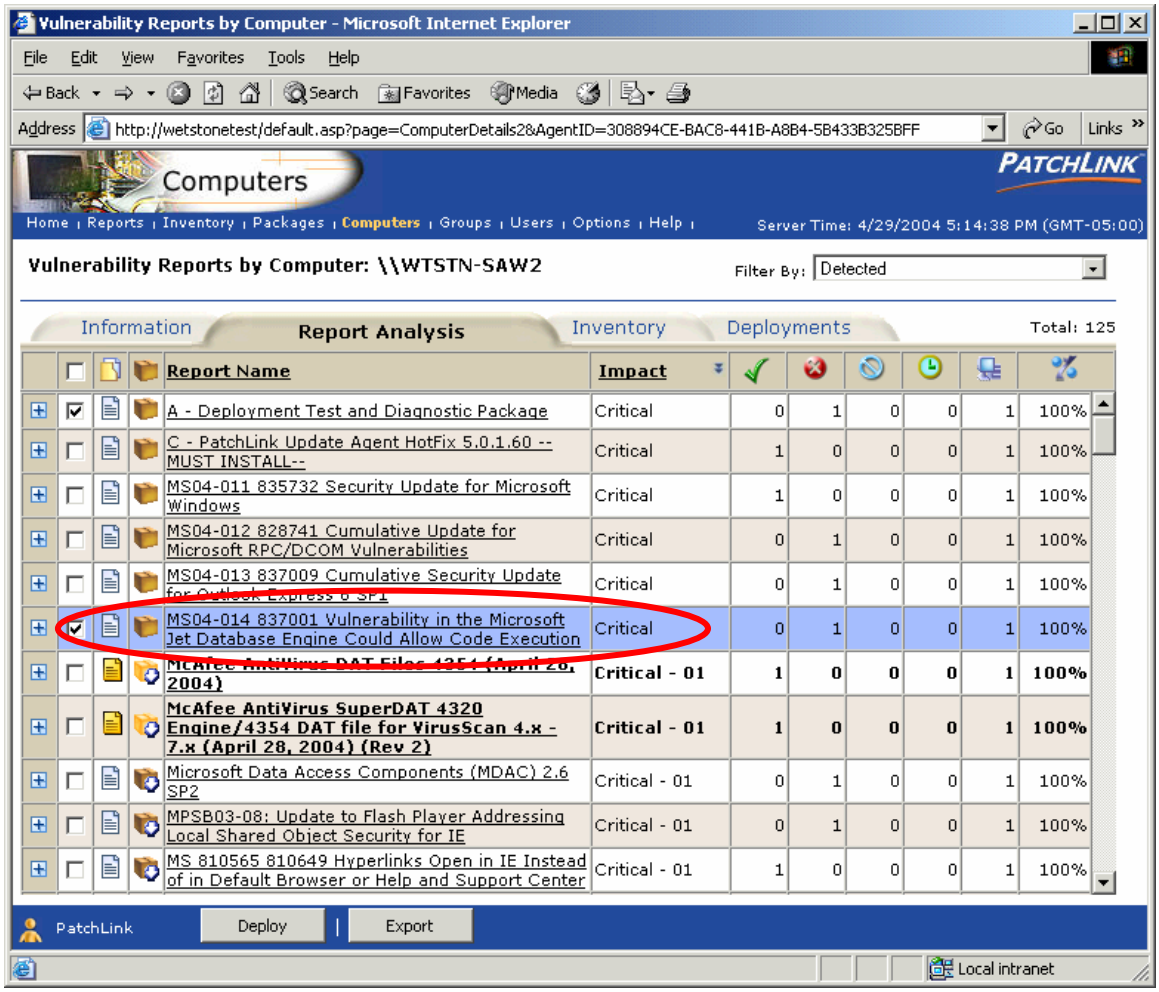


Network Benchmarking



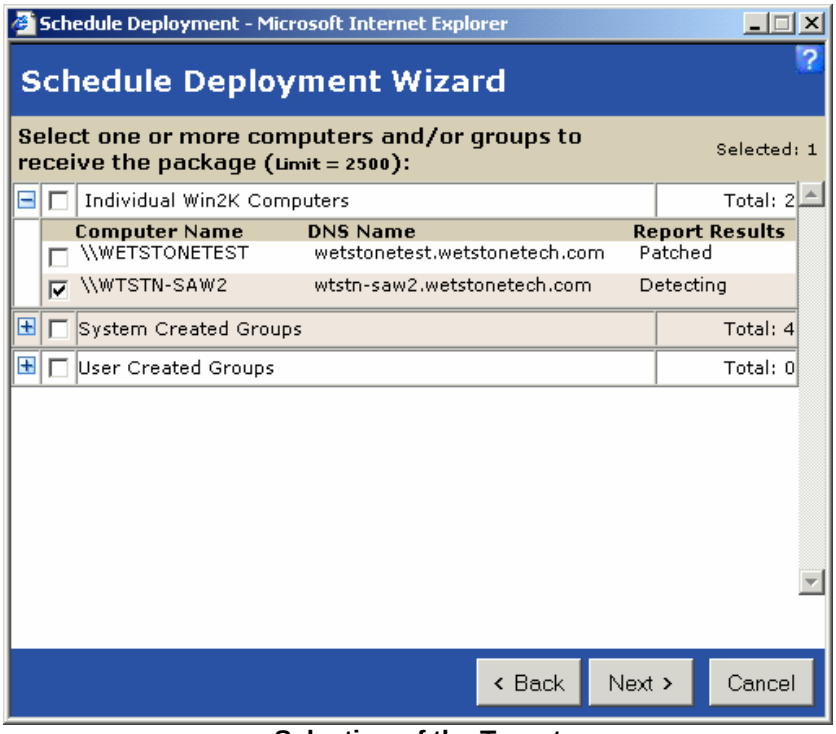
Harddisk Benchmarking

After reviewing the information gathered from our various resources, we were able to determine that our target machine had the basic operating system for this patch, the PatchLink update program recommended patching the system, the patch had not been previously applied according to even the lower-end interrogators, the overall health of the client machine was good, and benchmarking tests verified that the system, including the network, were more than satisfactory to produce a good patching result. Thus, we began deployment using the PatchLink tool.

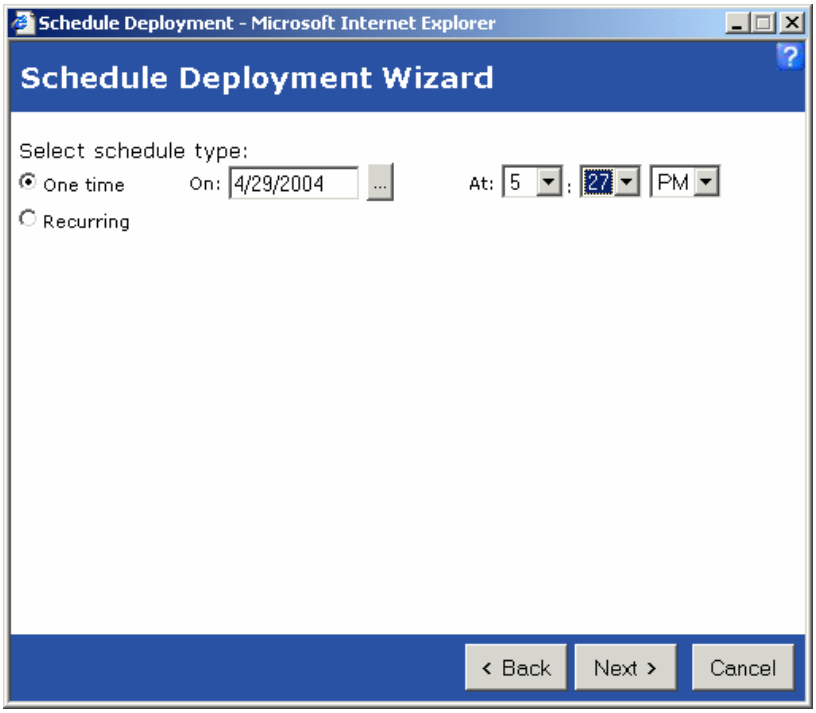


Patch selected for Deployment

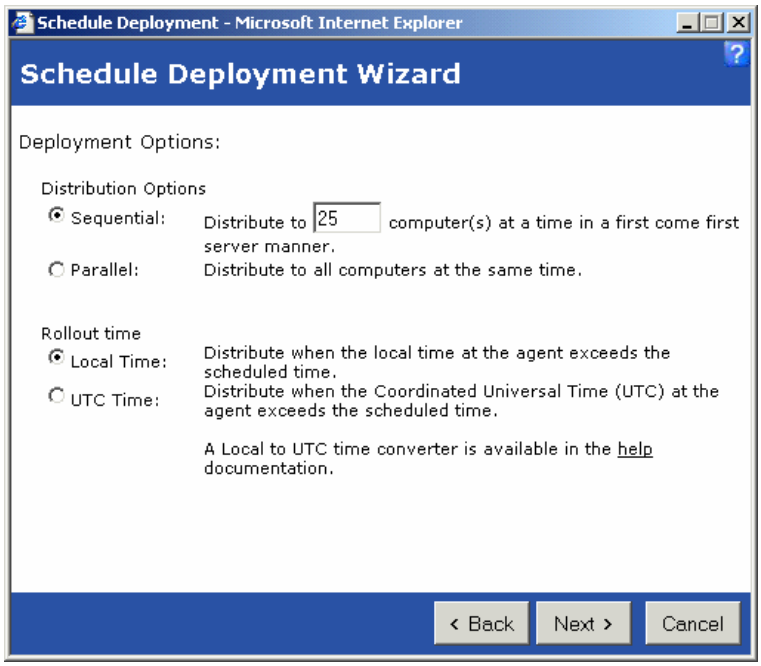
A wizard now appears. Following are the screenshots in the wizard for deployment of MS04-0014.



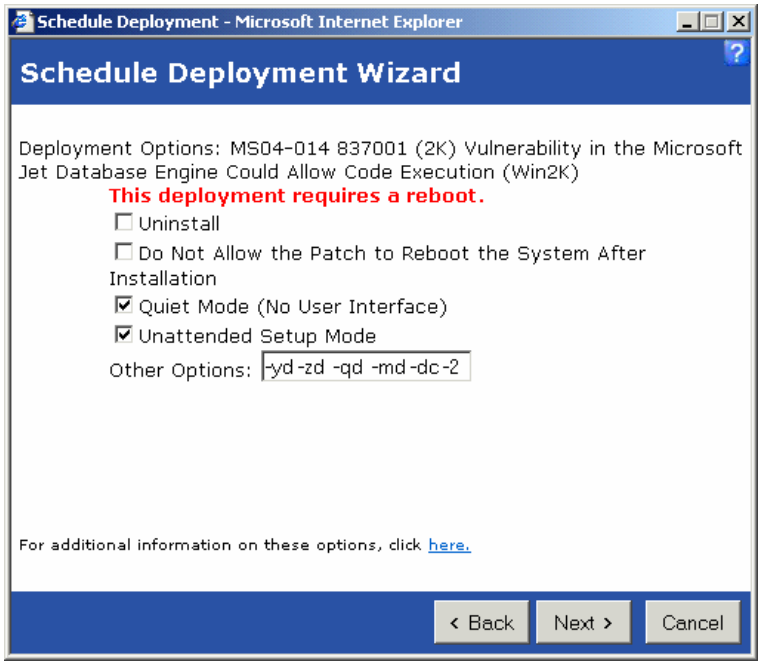
Selection of the Target



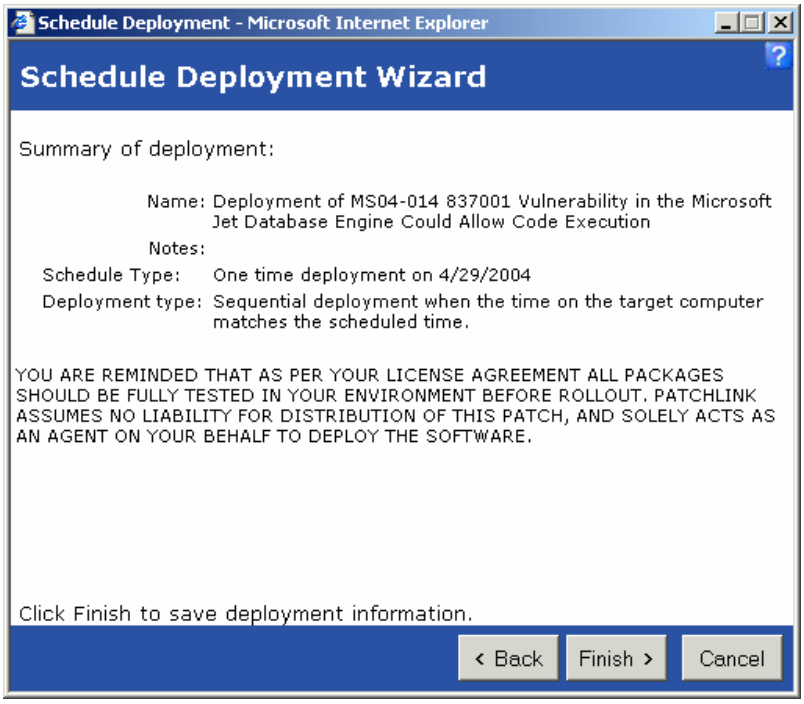
Time Schedule for Deployment



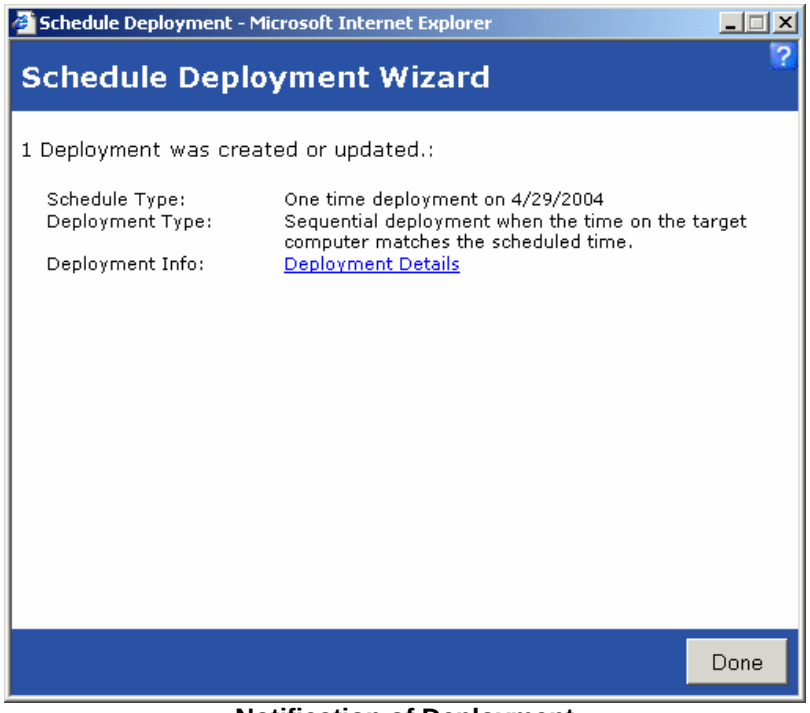
Options for Deployment



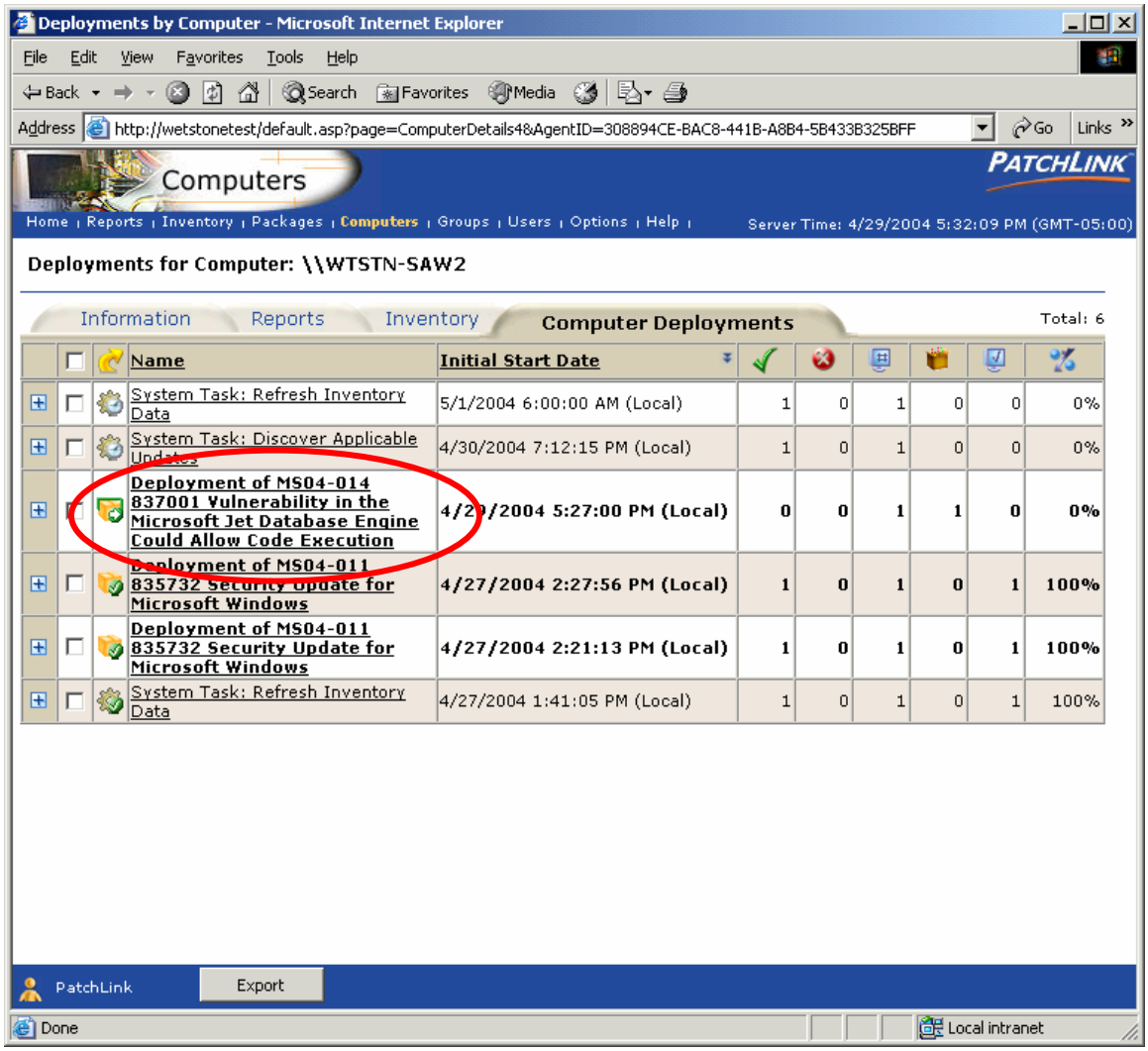
Mode of Installation – Reboot Information



Final Deployment Information

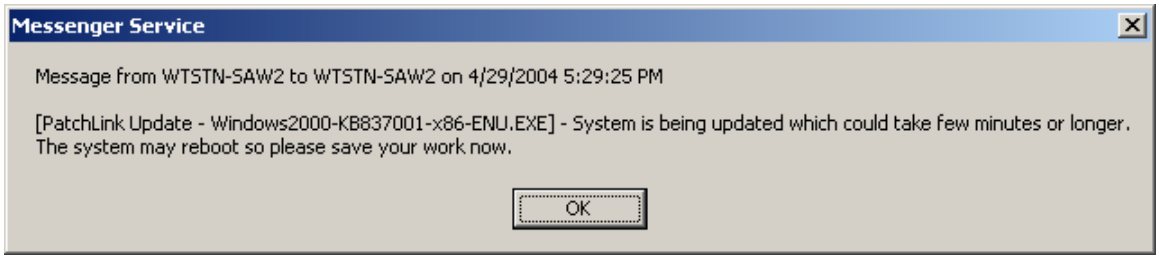


Notification of Deployment



Verification of Deployment

The next screenshot shows the messenger service popup sent from the PatchLink Server to the client.



The client machine successfully rebooted and continues to be fully operational after remote installation of MS04-014 from PatchLink Update.

2.4.4 *Create Matrix of Attributes that can be Extracted (Automated and through Q&A) About a Critical System*

After examining the tools and technologies that are available to System Administrators today and executing the two case studies described above, we have developed the table below to identify the Critical System Attributes that can be extracted from critical system installations today.

Table 7 Critical System Attribute Table

Attribute Name	Description of Attribute	Attribute Importance	Method of Retrieval
Operating System	The basic system upon which all operations are performed	This is the primary information that is needed before applying a patch	Manual Process or Automated Inerrogation Tool
Software inventory	List of all program packages installed on the patch candidate node	This is the most widely used and distributed attribute for a patch	Automated Interrogation Tool
Software version	The version number of installed applications	Some patches are only appropriate to specific versions of software	Automated Interrogation Tool
Hardware inventory	List of all internal devices installed on patch candidate node	In the case of a patch for a device (driver update) an inventory of the internal components needs to be available	Automated Interrogation Tool
Driver versions	A list of the drivers	To insure proper	Some interrogation tools

Attribute Name	Description of Attribute	Attribute Importance	Method of Retrieval
installed	controlling the devices from the hardware inventory list	patching, it needs to be determined that the proper patch is installed and if the patch can be applied out of sequence, e.g., install patch A and then patch C if patch B was not installed upon implementation of the device	provide such information. A manual inspection of the particular system would also provide this information
Available drive space	The amount of unused space contained on a hard drive	Many vendor patches are quite substantial in terms of shear size and need more drive space than may be available	Automated interrogation tools or querying the target machine through, for example, system information in Device Manager on a W2K machine
Physical memory	The number in Megabytes installed on the target machine. Possibly the memory module configuration would be helpful to have in the event an upgrade or replacement is necessary.	This is a monitoring tool to determine the health of the target system	Tools such as Belarc Advisor supply this information, as well as the BCM Advanced ToolBox.
Sufficiency of swap space	The amount of virtual memory in the target machine	This is another tool to monitor the health of a system. If the swap space is insufficient, operations such as patch application may halt	The BCM Advanced ToolBox (or one of the similar tools found at http://www.buildorbuy.org/downloads.html) will provide this information. A manual interrogation of a system will also give this data. For instance, this information in W2K is found in a Control Panel Applet

Attribute Name	Description of Attribute	Attribute Importance	Method of Retrieval
Health of machine	The general overall health of the patch candidate machine, including but not limited to, internal temperature, health of the hard drive (or boot sector), stress tests, memory test, and processor test.	It is important that the target machine be in a state that will receive a patch successfully	Any one of a number of diagnostic software applications available will provide this information, including BCM Advanced ToolBox
Benchmark information	More health information gathered to determine if the target machine is up to par for patching since the last round of patches. The benchmarking could be as compared to other brands of machines of like configuration, or it could be benchmark information on the same machine over time.	It is important that the target machine be in a state that will receive a patch successfully	This information can be retrieved from applications such as FreshDiagnose
List of deployed patches	A list of patches that were deployed to which applications, along with the date of deployment.	This is useful tracking information for installation of a current patch, or uninstallation of previous patches	This information may have to be gathered manually. FreshDiagnose does provide Microsoft patches only.
Function of patch candidate machine	The purpose of the target machine, whether it is a mission critical system, a client workstation, a web server, or anything in between.	This information is valuable in determining the criticality of applying a patch, or the scheduling of such application.	This is a manual determination by the administrator.
Backup System Availability	Does a backup hot or cold exist for this system?	This factor will be used to assess risk	Manual
How long will a rebuild of this system take if a failure occurs?	If the system should fail for any reason during upgrade the time frame of recovery wil be critical	This factor will be used to assess risk	Manual
Primary System Function	Understanding the primary function of the a candidate critical system will help us to determine if a patch is appropriate	This could be critical if depending upon other risk factors associated with the patch.	In most cases today this is information exists within the knowledge base of the System Administrators and IT

Attribute Name	Description of Attribute	Attribute Importance	Method of Retrieval
	for that machine. For example a machine that has the primary function as an e-mail server with no other services in operation may not require a pathc to be applied that is specific only to and Web Server Fault.		personnel. In some cases the CIO or CTO may have a network map that is properly updated.
Mission Critical Nature of the System	Critical questions as to the mission critical nature of the system. For example, does a backup system exist? What would be the impact on the mission if patching the system caused a catastrophic system failure? How long would recovery take under these conditions?	High	In most cases today this is information exists within the knowledge base of the System Administrators and IT personnel. In some cases the CIO or CTO may have an understanding of this. However, interdependency issues may be less clear or not intuitive. For example if an Network Time Protocol Server requires a patch due to a newly discovered NTP bug. The mission critical nature of such a system may be underestimated. If for example network time synchronization across multiple application servers is essential then losing syncronization servers may degrade or cause unexpected failures in these systems.

2.5 Using Scoring Models for Patch Risk Analysis

"When a vulnerability is discovered and a related patch and/or alternative workaround is released, the entity should consider the importance of the system to operations, the criticality of the vulnerability, and the risk of applying the patch. Since some patches can cause unexpected disruption to entities' systems, organizations may choose not to apply every patch, at least not immediately, even though it may be deemed critical by the software vendor that created it. The likelihood that the patch will disrupt the system is a key factor to consider, as is the criticality of the system or process that the patch affects."[xi]

"FedCIRC officials emphasize that although the contractor tests the security patches, these tests do not ensure that the patch can be successfully deployed in another environment; therefore, agencies still need to test the patch for compatibility with their own business processes and technology."[xii]

The final stage of our study is to analyze scoring models (e-commerce transactions, fraud and credit) to determine the adaptability of those models to one that is able to score risk associated with patch deployments. The advantage of examining scoring models for patch risk assessment intuitively makes good sense. First, scoring models for credit, e-commerce and fraud have one common theme that is to mitigate risk, thus relating directly to our objective here. Second, all the aforementioned risk models have some but not all the information necessary to make perfect risk assessment, therefore, they must rely on a multitude of methods to fill in the blanks and make probabilistic judgments. During this section we will define some of the basic scoring methods that we examined and illustrate corollaries to software patch deployment. In addition, we will assess the feasibility of adapting these models and make specific recommendations.

2.5.1 Scoring Model Overview

What is a scoring model? Depending on the type, a scoring model performs a mathematical calculation based on a set of input variables and produces a numerical value. In the case of e-commerce transactions scoring, typically the higher the score the higher the risk of fraud, in the case of credit scoring the higher the score the lower the risk.

During the past 20 years lenders in financial institutions have adopted new decision making models to accelerate loan decisions and to estimate both short and long term

credit risks. As far back as the 1970's scoring models have been used to evaluate the issuance of residential mortgages.

In addition, the use of these models has extended beyond simple credit risk models which indicates their flexibility. "The use of credit scoring technologies has expanded well beyond their original purpose of assessing credit risk. Today they are used for assessing the risk-adjusted profitability of account relationships, for establishing the initial and ongoing credit limits available to borrowers and for assisting in a range of activities in loan servicing, including fraud detection, delinquency intervention and loss mitigation. These diverse applications have played a major role in promoting the efficiency and expanding the scope of our credit delivery systems and allowing lenders to broaden the population they are willing and able to serve profitably"[xiii]

Based on this early work several commercial firms have developed a host of models for varying purposes. The following table summarizes some of the most popular models and their description. Please note the basic table below was created from the noted reference, however, we have mapped the associated scoring model to the potential adaptation to patch risk assessment. [xiv]

Table 8 Overview of Scoring Models

Scoring Model	Developer	Type	Description	Potential Patch Risk Adaptation
AdvanceBK	Fair Isaac	Bankruptcy Prediction	Model design optimized for bankruptcy also include non-bankrupt charge-off; using a combination of transaction, issuer supplied account performance data and 3rd party information	Vendor Risk associated with a vendor's patch releases history. This model would use historical information regarding a vendors past performance in successful patch releases.

Scoring Model	Developer	Type	Description	Potential Patch Risk Adaptation
DELPHI	Experian	Bankruptcy Prediction	Predicts the likelihood of bankruptcy within the next 12 months	
Credit Union Risk Model	Experian	Industry Specific Risk	Predicts the likelihood of seriously delinquent or derogatory credit behavior on a credit union account over the next 24 months (including revolving, installment, auto and mortgage accounts).	
Application Risk Model	Fair Isaac	Application Risk	Application risk models are based on a national pool of lending data and designed to give consumer lenders a cost-effective means to assess credit risk for a variety of portfolios, such as revolving, direct, indirect, and home equity line of credit loans. Empirically developed specifically for use in credit origination decisions	Patch risk assessment that would examine the information surrounding a specific patch and assess the general risk of applying it. The type of patch (i.e. operating system, application, security vulnerability, bug fix etc.) would be examined as well.

Scoring Model	Developer	Type	Description	Potential Patch Risk Adaptation
ASSIST® 2.0	Fair Isaac	Insurance Risk	Rank orders applicants and policy holders by risk in terms of likely relative loss ratio.	This type of model could be applied to assess the potential loss that would occur if a critical system was damaged during a patch update. This type of model could also be used to develop tradeoff strategies to help assess the tradeoffs associated with patching a system. This is not to say the patch would not be applied, but rather would determine when the patch should be applied, what additional testing and evaluation should be accomplished and what rollback capabilities should be in place prior to patch deployment.
Authentication Solutions Level One Score	Experian	Fraud Verification	Verifies consumer information including name, Social Security number and telephone number	Models of this type could be applied to assess the risk associated with a malicious patch. Current methods exist today (digital hash, signature and timestamps) that prove the authenticity of the patch, in other words whether the patch is legitimate. This does NOT prove that the patch
Authentication Solutions Level Two Score	Experian	Fraud Verification	Verifies the likelihood that the correct consumer supplied credit application information	

Scoring Model	Developer	Type	Description	Potential Patch Risk Adaptation
Fraud Detect Model	Advanced Software Applications	Identify Fraud	Verification tool that evaluates applications for inconsistencies in information provided by consumer, Likelihood of using fraudulent information	is safe and free of malicious code or timebombs. In the Critical But Missing Patch Information, we suggest that vendors should provide critical information regarding the pedigree of the patch (personnel, location work was performed, supervision, code walk-throughs etc.) This information could be utilized to assess fraud risk using models like this one.
Visa Issuer Fraud Detection	Visa	Fraud Detection	Predicts based on authorization patterns, merchant profiles and cardholder spending profiles (hybrid modeling technology) Automatically refreshed using recent world fraud trends. Looks outward 24 months.	
Pinnacle	Fair Isaac	Risk	Rank orders consumers according to the likelihood of future default on credit obligations. The next generation FICO score provides more refined assessment across the entire credit risk spectrum looking outward for	This model may provide additional insight into vendor risk based on additional factors looking forward. Examining future factors is certainly beyond the scope of this study, however, examining economic conditions, market pressures, competitive forces, law suits etc. against certain vendors may raise risks

Scoring Model	Developer	Type	Description	Potential Patch Risk Adaptation
			24 months	associated with future patch releases. In the credit and financial world these rating are typically used to intervene or work with the business or consumer to avert or preempt future problems. Possibly this type of model could be applied in a similar manner and give critical infrastructure managers advance warning of the road ahead with certain vendors and take preemptive actions to avert possible problems.
SPECTRUM®	Scoring Solutions Inc.	Risk	A risk model for the wireless communications industry. Predicts the likelihood of a customer becoming seriously delinquent or result in loss within the next 6 months.	

E-commerce transaction processing offers additional scoring models for evaluating the probability of fraud during Internet transactions. Forrester research forecast U.S. online retail sales this year will close at $95.7 billion, climbing to $229.9 billion in 2008. Next year's sales are projected at $122.6 billion, with $149.2 billion, $176.8 billion and $204.3 billion forecast for 2005, 2006 and 2007, respectively. This dramatic increase has fueled the development of new fraud scoring models and methods for Internet merchants to filter out the good from the bad. The scoring models developed for this purpose are essentially risk mitigation strategies. Some are quite simple, while others are complex and include extensive backend databases and processing.

One of the simplest methods employed by online merchants to curb fraudulent transactions is AVS or Address Verification Service. According to CyberSource Corporation, one of the leading commercial companies offering AVS service, "AVS is currently beneficial for supporting the screening of purchases made by US consumers. The AVS check, designed to support mail order and telephone order businesses, is usually run in conjunction with the bank card authorization request. AVS performs an additional check, beyond verifying funds and credit card status, to insure that elements of the address supplied by the purchaser match those on record with the issuing bank. The following is a summary of responses merchants can receive from an AVS check:

Response	Description
AVS=MATCH	The first five digits of the street address, the zip code, and credit card number match those on record at the bank.
AVS=PARTIAL MATCH	There is a partial match (e.g., street matches but not zip code, or zip code matches but not street).
AVS=UNAVAILABLE	The system cannot provide a response. This result is returned if the system is down or the purchaser does not reside in the United States (AVS is only available for US residents).
AVS=NON-MATCH	There is no match between the data elements

While most merchants will not accept orders involving issuer declines or AVS=NONMATCH, the automated nature of an online transaction requires merchants to implement policies and processes that can handle instances where the card has been approved, but other data to validate a transaction is questionable. Such instances include cases where the response is "Issuer Approved" and AVS = PARTIAL MATCH or UNAVAILABLE (e.g., the purchaser's bank approved the transaction, but it's not clear whether the transaction is valid)."[xv]

Turning to patch risk management, how might a simple model like this be applied? The following table illustrates a mapping of this model to patch management.

Table 9 Patch Management

Response	Description
Patch Applies	The patch directly applies to the target critical system. The operating system or application in question is critical to functional operation of the critical system. The patch addresses a security vulnerability that could cause damage or degradation of service to the critical system if it is not applied. In this category a policy may be in place that states patches of this classification must be applied in 72 hours.
Patch Partially Applies	The patch is appropriate for this type of system; however, the services that are affected by the patch are not in use by the system. For example the patch applies to all Microsoft® Windows systems and the patch addresses vulnerability in the SQL Server. The system under consideration is not running SQL Server today. However, the patch replaces several system DLL's that are in use by this critical system or there is a plan to integrate an SQL Server into this implementation in the next six months. Patches that fit this category may be scheduled by policy to be deployed during normal system upgrades cycles.
Patch Does Not Apply	In this case the patch is clearly not appropriate for the critical system in question. One reason could be that the patch applies to an operating system version of LINUX that we are not running or the patch is so isolated (fixes security vulnerability in Microsoft Exchange Server version 2.46) and affects no other system files and we are not using or never intend to use Microsoft Exchange on this system. Patches in this category can be so labeled by policy so that when local vulnerability scans are run against the critical infrastructure that identifies missing patches the missing patches can be quickly reconciled.
Patch Information unavailable	In this case not enough information is available either for the critical information system or from the vendor regarding the proposed software patch.

Therefore even a simple model like AVS could be used as a basis for categorizing, performing triage on patches and communicating about their status or deployment strategies.

In addition to the basic model described above, most AVS vendors also provide advanced services that provide additional information and warnings regarding potentially fraudulent transactions. The following table represents most of these additional variables[xvi]

Table 10 Vendor Services

Code Title	Code	Description
Excessive Address Change	A	the customer has 2 or more billing address changes in the last (timeframe)
Bin Check	B	the bin check failed
High Count of Unique Credit Cards	C	the customer has more than "X" credit cards in the last (timeframe)
Domain (Host)	D	the customer has a risky domain (IP) or e-mail address
Fraud List Flag	F	a previous merchant has incurred a chargeback with no return of product
Geo-location Inconsistency	G	the correlation between the customer's e-mail or IP address (and possibly other factors) and stated billing address is suspicious
Name Change	H	the customer using this card has 2 or more name changes in the last (timeframe)
Internet Inconsistency	I	the correlation between the customer's phone number, billing address, shipping address and other factors has been determined to be suspicious
Nonsensical Input (gibberish)	N	the customer input contains highly unbelievable data in the customer name and address fields
Obscenities	O	the customer input obscene words in th order form
Time Hedge	T	the customer attemps a purchase outside the expected hours for purchase of the item
Unverifieable Address	U	the bill_to_address or ship_to_address is not verifiable
Velocity	V	this card has been more than "X" times in the last "Y" minutes
Warning	W	there is only a partial address match
Unclear Request	Z	the information in the request contains an unusual or unexpected value; examine the request carefully for abnormalities in the order
Time	displays 00:00 format, the order time in the customer's local time	
Host Severity	numeric 0-5 format; the risk associated with the customer's e-mail domain	

Several interesting factors could be equated to software patch risk analysis. For example:

Bin Check : This could be equated to validation error of the hash, digital signature or timestamp provided by the vendor for this patch.

Fraud List Flag: This could be equated to reports from other system administrators regarding problems found with the application of the patch in question.

Geo Location Inconsistency: This warning could be associated with risks based on where or by whom the patch was created. For example, the vendors could provide information that the patch was modified by a Russian programming group that is part of

the company and no code walk-through was conducted to verify that malicious code wasn't intentionally inserted. This could provide another indicator of risk.

Internet Inconsistency: Validating the consistency of the vendors claims regarding what this particular patch contains. For example, the vendor suggests that this patch relates to a buffer overflow problem in a protocol stack component of the system. Under further analysis, you find that a memory management, process management and display subsystem are included in the patch. Therefore a clear inconsistency exists in the vendor claim and the content of the patch. Several checks for consistency and completeness could be conceived that not only are directed at subsystem components but also could be in the consistency of the size of the patch (number of files or code changes made) vs. the purported reason for the patch.

Obscenities: How could obscenities possibly be mapped to patch risk management? Obviously, malicious code scans for the proposed patch are routine steps for most IT departments. If the software patch is an install package, they extract out the individual files within the patch (by uncompressing the tar, zip, or cab files) and run the latest virus signatures against the proposed patch. This provides one more level of protection against contaminating and otherwise secure environment.

Velocity: The measure of velocity of software patches could be extremely valuable to determine risk of a specific patch and also help in determining course of action. Today, several companies have adopted the 48 hour wait and see strategy for patch deployment, whereby they wait 48 hours before applying a patch. During that cooling down period they monitor the Internet and user groups for both problem reports and successful confirmation of the fix.

2.5.2 The Math Behind the Models

It is certainly out of scope for this report to include the details of the scoring models employed for e-commerce, fraud and credit systems. However, a summary of the broad set of approaches utilized may prove to be informational. The following section was compiled from an article written by Fair Isaac that outlines the various models that they use in scoring.

Table 11 Fair Isaac Scoring Models

Scoring Method	Application	Strength	Weakness
Discriminant Analysis:The goal in discriminant analysis is usually two-fold: 1.Segment or separate individuals into two or more previously defined groups. Classify a new individual into one of the groups A rule or "discriminant function" is developed based on measurements (variables) associated with each of a sample of individuals from two or more populations. As in *regression*, the general approach is to construct, in some optimal way, a *linear combination* of measurements or predictor variables which will best distinguish (discriminate) between the groups. The model is in the form of multiple formula, each corresponding to one group9. A new individual can then be assigned or classified into the correct population based on the highest value of the linear combinations (scores) from among the discriminant functions for that particular individual.	Often used in marketing (e.g., to distinguish purchasers of a new product from non-purchasers, to identify low/medium/high response groups). Also used for developing credit risk models. Successful applications of expert systems and case-based reasoning can be found in the areas of personnel policies, maintenance rules, financial planning, and medical diagnosis. In the risk management world, expert systems have been applied in areas where data were not readily available, namely mortgage application and small business loan processing.	- Can separate and classify individuals into multiple groups. - The idea of scoring an individual and use of a cutoff is inherent in this methodology. Hence it can be easily perceived as the "right tool" for credit scoring. - Can model multiple outcomes.	- Assumes that the predictor variables are distributed as multivariate normal (having a combined distribution that is normal in multiple dimensions—this results in some elegant simplifications on which discriminant analysis relies). This assumption is usually violated in our typical scoring applications. Although the technique is somewhat robust with respect to minor violations of the assumption, serious violations will often result in unreliable estimates. - If stepwise discriminant analysis is used, the problems associated with variable selection procedures are present. The "best" subset selected for a given data set may perform poorly in future samples. - When some or all of the independent variables are very highly correlated (i.e., a situation often termed

			multicollinearity), the procedure could select an unreasonable set of variables as optimal. In fact, in situations of multicollinearity, estimates of regression coefficients from sample to sample fluctuate markedly.
Expert systems, often called knowledge-based systems or rule-based systems, are computer software applications that capture the knowledge of a human expert and make decisions based on this "knowledge base." The knowledge base is represented by a set of IF-THEN rules. This set of rules is determined in one of two ways. The traditional approach is to have a "knowledge engineer" work through an interview process during which the engineer extracts the knowledge from the expert. Alternatively, if a database of cases along with the expert's decision is available, the knowledge engineer can induce a set of rules from this database using a rule induction technique such as trees. Regardless of the technique, the result is a knowledge base of IF-THEN rules that are programmed into software. Once the software is programmed, the expert system uses its "inference engine" to access the knowledge base, sort	Successful applications of expert systems and case-based reasoning can be found in the areas of personnel policies, maintenance rules, financial planning, and medical diagnosis. In the risk management world, expert systems have been applied in areas where data were not readily available, namely mortgage application and small business loan processing.	- It is very appealing to clone the corporate experts. Many contract signers would regard themselves as experts. - Expert systems do not require data. - They are better than nothing in terms of supplying management control and a way of exerting some consistency in the decision making process.	- The knowledge extraction process is very difficult and time consuming. - Poorly engineered solutions can be a nightmare, or impossible, to maintain. Changes in the thinking of the expertise can affect the whole structure of the decision system. - When adequate data is available for formal analysis they are inferior global alternatives to data-driven solutions.

through the set of rules, and make decisions on new cases, allowing the human expert to focus his or her attention on the more difficult decisions			
Genetic algorithms (GAs) are a class of *optimization algorithms* inspired by population genetics and the Darwinian principle of natural selection, commonly referred to as "the survival of the fittest." Given an *objective function*, the typical GA begins with a random population (generation) of solutions (chromosomes). Each solution is represented by a sequence of characters (genes) each having certain values (alleles). By mating and mutating the best solutions (as measured by some fitness value), the GA produces a new population of improved solutions (offspring). The average fitness of the population, as well as the fitness of the best solutions, improves at each generation. This process continues until the GA has determined an acceptable solution to the problem (as determined by the developer).	Flexible encoding allows genetic algorithms to be applied to a diverse set of problems in biology, computer science, engineering and operations research, image processing and pattern recognition, and the social sciences. Their highly parallel search mechanism makes them suitable for high-dimensional, highly non-linear, non-smooth objective functions that other optimization techniques find difficult to solve. In general, however, genetic algorithms will generally take longer to converge than other techniques, and as with other optimization techniques, are not guaranteed to find the globally optimum solution.	- General-purpose technique that is applicable to a variety of problems. -Generally finds a good solution.	- Not guaranteed to find the best solution. - Computationally intensive.
Graphical Decision Models: Graphical paradigms play an important role in modeling and structuring decision problems11. The two most commonly used graphs to display decision models are influence diagrams and decision trees. The following types of nodes are	Influence diagrams are a powerful tool in modeling decision problems, because they allow for the specification and visualization of the structure of fairly complex problems in a compact graph that conveys explicitly the assumed dependence, or independence, among	- Allow for the visualization of complex problems in a compact way, particularly the dependence structure among variables. - Effectively	- Detail behind each node in the graph is not readily apparent. - Typically unable to capture the asymmetric structure of a decision problem14.

used in both types of graphs: - Decision nodes, drawn as rectangles, represent decisions. - Chance nodes, drawn as ovals, represent uncertain events. - Consequence or value nodes, drawn as rounded rectangles or diamonds, represent consequences.	variables, the sequence of decisions, and the flow of information to the decision maker. They are most effective in the early stages of modeling an unstructured problem, when data and other details are unavailable, as a communication tool between a decision analyst and a decision maker. In conjunction with *sensitivity analysis*, they allow the determination of what matters in a problem and what does not, and thus the construction of tractable models that allow insight into the problem and its solution.	communicate the relationships between variables and the sequence of decisions. - Serve as a formal framework for Bayesian inference and learning.	
Decision Trees: In contrast to influence diagrams, decision trees explicitly show any asymmetry in the structure of a decision problem. They also show the functional and numerical details for each node on the corresponding branches. Each branch emanating from a decision node corresponds to an alternative and each branch emanating from a chance node corresponds to a possible outcome. ■ When there is no Response, the immediate realization of Profit15 is the end of this scenario; other events, like Income and Performance, are never realized, and the Credit Limit decision never gets to be made; ■ When the customer applies but the decision maker decides to not grant credit, the immediate realization of Profit16 is similarly the end of this scenario.	Decision trees preceded influence diagrams by many years and are still indispensable when a highly asymmetric decision problem needs to be structured and modeled graphically. They are useful when used in conjunction with influence diagrams.	- Details associated with each node are readily apparent in the graph - Asymmetric structure is readily displayed.	- Decision trees become unwieldy for decision problems with even a moderate number of variables or a few stages; - Conditional dependence and independence among variables are not readily apparent in the graph.

Linear and Non Linear programming (LP) and non-linear programming (NLP) are two widely utilized techniques to minimize (or maximize) an *objective function* subject to constraints. Both LPs and NLPs are subclasses of the field called mathematical programming, which originated in the 1940s, when the term 'programming' was still synonymous with scheduling or planning. Mathematical programming solutions are utilized when there is no closed, algebraic solution for determining the optimum value of the objective function, or when the derivation of an algebraic solution requires more time and effort than a mathematical programming technique.	Linear programming and non-linear programming are utilized widely to solve prediction and decision problems in the areas of finance, operations management, economics and the physical sciences. NLP techniques are often hidden within commonly used multivariate statistical software programs (e.g., maximum likelihood estimation for log-linear models) and in decision optimization software.	- Many techniques are available, so if one does not work for a particular problem, another might. - LPs and NLPs handle a wide variety of objective functions and constraints. - Mathematical programming is a well-researched area, so that guidance is available in the literature to help determine appropriate techniques for particular problems.	- There is seldom a guarantee that a particular technique will converge to a solution for a particular problem, nor that the solution converged to will be a global minimum. - For some problems, much of the work is in the correct specification of the objective function. - Because these methods are iterative, they can be computationally intense and require long execution times.
Link Analysis: Computer-based link analysis is a set of techniques for exploring associations among large numbers of objects of different types. These methods have proven crucial in assisting human investigators in comprehending complex webs of evidence and drawing conclusions that are not apparent from any single piece of information. These methods are equally useful for creating variables that can be combined with structured data sources to improve automated decision making processes.	Link analysis is increasingly used in law enforcement investigations, detecting terrorist threats, fraud detection, detecting money laundering, telecommunications network analysis, classifying Web pages, analyzing transportation routes, pharmaceuticals research, epidemiology, detecting nuclear proliferation, and a host of other specialized applications. For example, in the case of money laundering, the entities might include people, bank accounts and businesses, and the transactions might include wire transfers, checks, and cash	- Link analysis often makes information accessible that is not apparent from any single data record.	- Link analysis is as endeavor.

	deposits. Exploring relationships among these different objects helps expose networks of activity, both legal and illegal.		
Log-linear models provide a systematic approach to the analysis and modeling of the observed cell frequency of occurrence in a cross-tabulation. Developed purely for understanding the structure and modeling of *categorical* data.	Log-linear models are most frequently encountered in the social sciences, where the need to understand relationships between categorical data is often required. Marketers have used log-linear models for response modeling, with trees as a pre-processor to reduce the number of variables. Fair Isaac's proprietary variable investigation tool, ADVISE, incorporates log-linear modeling to automatically identify potential *interactions* between pair-wise combinations of candidate predictor variables.	- Provides methods for analyzing categorical data that are analogous to correlation and regression analyses of continuous data. - One of the more effective approaches for detecting low-dimensionality interactions between variables. - One of the more effective approaches for detecting low-dimensionality interactions between variables. - Makes no assumptions about the distribution of the predictor data. - Appealing as a segmentation tool, as it identifies unique segments of data. - Provides an interpretation of the direction and magnitude of relationships in multi-dimensional tables.	- Data get sparse quickly as dimensionality increases. - Model is usually limited to low level of dimensionality, unless a very large sample of data is available. To be effective, this technique needs to be combined with a variable reduction pre-processor. - Requires data to be categorical.
Neural Networks: A neural network26 (NN) is an	Neural networks, and multilayer perceptron neural	- Model non-linear, non-additive	- Provide little data insight and are

information processing structure that transforms a set of inputs into a set of outputs. The manner in which a NN performs this transformation is inspired by researchers' understanding of how the human brain and nervous system process information. More specifically, a NN is a collection of simple processing units linked via directed, weighted interconnections Each processing unit receives a number of inputs from the outside world and/or other processing units, weights these inputs based on the weights of the corresponding interconnections, combines these weighted inputs, produces an output based on this combined input, and passes this output to other processing units via the appropriate weighted interconnections. Mathematically, this process can be represented by a function that maps the set of inputs to a set of outputs. In general, this function is *non-additive* and *nonlinear.*	networks in particular, have been used to address a variety of problems, a few of which are listed below: - Optical character recognition - Industrial adaptive control systems and robotics - Image compression - Medical diagnosis based on a set of symptoms - Statistical modeling	relationships in data - Handle both continuous and categorical predictors and outcomes - Handle multiple outcomes in a single model - Are not a proprietary technology (i.e., are readily available as software)	difficult to interpret - Can overfit the development data if used naively29 - The solution may be sensitive to the starting point due to the possibility of multiple locally optimal solutions
Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant *features* or attributes of the data from a background of irrelevant detail. The historically most frequent areas of application are in spatial pattern recognition—3-D image processing, character and voice recognition, and in temporal pattern recognition—weather forecasting and	Pattern recognition techniques are often used for image processing, character and voice recognition, as well as weather forecasting and financial time series forecasting. Applications continue to expand with recent examples in the area of credit risk, marketing and fraud detection model development. Descriptive modeling of web site behavior built by analyzing click-stream data is	- Can increase the predictive power of classifiers substantially by finding valuable new patterns. - Automated search capabilities inherent in most pattern recognition techniques can leverage analyst time and hasten the	- Patterns discovered might be spurious or not representative of future cases. Sample tuning can be an issue with some pattern recognition techniques - Definition of a "valuable" pattern might be unique to a

financial time series forecasting.	another area with success in the pattern recognition field.	learning process for new data sources or classification problems. - Wide field applicable to many problems across many different industries.	particular problem. Borrowing pattern recognition techniques from a different problem without consideration can produce meaningless features and classifiers.
Regression is a family of prediction modeling techniques. When “regression” is mentioned, care must be taken to understand which technique is being discussed to avoid misunderstanding. The goal of regression, as in many competing techniques, is to model the relationship between *predictor variables* and the desired *outcome variables* so that in the future, when the outcome variable is unknown, it can be estimated or predicted.	Regression is probably the most widely used technique for building models involving continuous outcome variables.	- Easy to interpret. - Widely used, well documented. - Can be a mixed model of continuous and categorical predictor variables. - Allows for a wide range of statistical diagnostics and significance tests.	Regression cannot elegantly handle missing values on a variable-by-variable basis. Data must be lost, or some assumption made about the missing data to give it a value. - Score weight patterns for categorical data cannot be made palatable. - The model assumes fixed increments/decrements in the score values for variables on an interval scale. - May not capture, or at least make readily apparent, interactions in data. - Categorical variables may have to be represented by dummy variables, i.e., multiple variables which represent the absence or presence of each component attribute in the predictor variable.

3 Conclusions

The goal of this brief effort was to determine the feasibility of developing a process that verifies if critical information system software patches behave as intended and introduce only the specific functionality identified for the patch. Based on our research, examination and experimentation, we have not only determined that it would be feasible to develop such a process, but also that this process and associated technology and standards are desperately needed.

These are our major findings based on the work performed under this study.

1. The number and frequency of software patches continues to escalate and inundate IT departments with yet another seemingly impossible task to deal with.

2. Software vendors continue to release and re-release patches. In many cases the patches have either failed to completely address the security vulnerability that they were intended to, or they cause other undesirable side effects.

3. Not all vendors deploy the same, and no standard exists for deployment of patches.

4. Patches vary widely – even when distributed by the same vendor for the same product family. This lack of standardization calls for handling patches on a case by case basis.

5. The release of patches by vendors (in most cases) are a direct result of the discovery of a vulnerability that either is, or is threatening to be, used by our adversaries to attack our information infrastructures. Based on this, many patches need to be developed, tested and deployed rapidly before attacks can be carried out. This rapid reactive model is creating a potentially dangerous situation.

6. Software vendors (in most cases) are recommending patches be applied ASAP to avert potential attacks.

7. Inadequate information is being provided by software vendors regarding the content of the patch distribution (i.e. the location of the flaw or the software components that are impacted by the patch), the genesis of the software flaw, the timeframe of the security flaw introduction, the who / what / when / how the patch was constructed or tested, the specific risks associated with applying the patches, or even simple metrics

regarding the amount of changes that the patch employs including size, files added, deleted, modified etc.

8. IT departments in many cases, are forced into “blind adoption” of patches due to the double edge sword of urgency to deploy patches to counter attacks and lack of “critical but missing information”. This situation forces them in many cases to play a real-life game of Russian roulette.

9. Many vendors, such as Patch Link, Novell, and IBM, now offer solutions that assist in the automatic detection and deployment of new patches to “appropriate” systems. However, they fall short in their ability to assess the risk of deploying such patches. For this the skill, expertise and ingenuity of those responsible are the only response. These people have come up with policies like “wait and see” that delay the introduction of patches only after waiting 24-48 hours to see how others have fared. Those with larger budgets or smaller infrastructures can employ the patches on backup systems first and then bring the backup systems online, if things work well, and only then apply the patches to the live systems.

10. Scoring models for fraud, credit and e-commerce transactions have evolved rapidly and offer both methods and techniques that offer promise to help in automatically assessing the risk of patch deployments. However, much work must be done to adapt these models.

4 Recommendations

Based on the work performed under this effort, we have developed specific recommendations that we feel will ultimately reduce the risk of patch deployments.

1. Expansion and Standardization of Patch Information

We recommend research to examine the specific information necessary by critical systems stakeholders when a patch is released. We recommend that an aggressive attempt be made to encourage software vendors to participate and cooperate in such a standardization effort. We would suggest that a standards body such as ANSI or OASIS be considered to host and develop such a standard. Furthermore we would recommend that vendors be required to provide this information for patches by including the requirements in all future procurements, much like the Y2K requirements exist today.

2. Automated Patch Assessment

We recommend the development of an automated patch assessment software system. The system would be designed to take inputs from the vendor (vendor claims regarding a patch deployment). The automated system would analyze the proposed patch system. Critical outputs would be:

1. Validate the claim of the vendor

2. Validate the integrity, authenticity and pedigree of the proposed patch

3. Scan the modified code for malicious code

4. Identify the subsystems affected by the patch through code analysis or other methods.

5. Identify risk factors based on the changes that the patch will make on the target system. For example changes made to process or memory management or security components may be deemed more risky than those made to esoteric features of an application.

3. Automated Critical System Interrogation Regarding Patch Deployment

We recommend research and the development of a critical information systems interrogation tool that would map a proposed patch to a target system. The proposed tool would interrogate and monitor a critical system to assess the critical software paths utilized by that system, the utilization of the system, and the mission critical nature of the system. A proposed patch would then be presented to the tool (mapping the software subsystems impacted by the patch). The system would then analyze the mapping and visualize the potential impacts of the patch on the target system, make recommendation for testing, and point out high risk areas and interconnectivity issues.

4. Automated Critical Information System Patch Deployment Scoring Model

We recommend research and the development of a patch risk scoring model that would analyze a proposed patch against a target critical system. The scoring model would leverage the work previously discussed in the areas of credit, fraud and e-commerce transaction scoring as a starting point. We would highly recommend a partnership with financial scoring companies to help rapidly develop such a scoring system. The ultimate goal would be for critical system organizations to request a score from such an organization for a specific patch to be applied to a specific target infrastructure. The scoring system would provide a base score and recommendations for deployment.

Recommendations would include deployment timeframe, recommended precautions, backup and rollback capabilities, pre-deployment testing and post deployment monitoring activities.

5. Automated Failsafe Patch Deployment

Finally, we recommend research and the development of a comprehensive patch roll back system. This system would allow for the automatic rollback of any installed or manually entered patches that fail. In addition, this system would roll back any data that was subsequently damaged or modified through the installation or subsequent actions caused by the patch deployment.

In conclusion, the difficulties that critical infrastructure stakeholders face regarding patch deployment will continue to increase as the complexity of software systems expand, the attacks on our critical infrastructures intensify and our reliance on these systems multiplies. It is certain for the foreseeable future the number and frequency of patch releases from major software vendors will continue to rise. These patches need to be applied and delivered to critical systems in a safe, comprehensive and rapid manner. In order to accomplish this, it is clear that we must immediately research and develop new methods and techniques to assist those that are charged with the management, operation and defense of these critical systems.

5 References

[i] "Effective Patch Management is Critical to Mitigating Software Vulnerabilities", Statement of Robert F. Dacey, GAO Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform September 10, 2003

[ii] IBID

[iii] "Windows Users Knocked off the Net" Associated Press, Wired News May 27, 2003 http://www.wired.com/news/technology/0,1282,59006,00.html

[iv] "Uprooting Software Defects at the Source" Hallem S., Park D., Engler D, ACM Queue vol. 1, no. 8 - November 2003

[v] IBID

vi "A Taxonomy of Computer Program Security Flaws", Landwehr Carl, Bull Allan, Mcdermott John, Choi William, ACM Computing Surveys, Vol 26, No. 3, September 1994

[vii] IBID

[viii] IBID

[ix] IBID

[x] "Effective Patch Management is Critical to Mitigating Software Vulnerabilities", Statement of Robert F. Dacey, GAO Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform September 10, 2003

[xi] "Effective Patch Management is Critical to Mitigating Software Vulnerabilities", Statement of Robert F. Dacey, GAO Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental
Relations, and the Census, House Committee on Government Reform September 10, 2003

[xii] IBID

[xiii] http://federalreserve.gov;

[xiv] "Credit Scoring for Risk Managers the handbook for lenders" Elizabeth Mays, South-Western 2004 ISBN 0-324-20054-4 (note the table has been enhanced under this effort to include additional information not included in the original text)

[xv] "Managing Risk on the Net White Paper What Internet Merchants Need to Know", CyberSouce® Inc. http://www.cybersource.com/resources/collateral/pdf/ifs_wp111500.pdf

[xvi] IBID